이학석사 학위논문

데이터베이스 은닉행위 식별을 위한 System Artifacts 조사 기법에 대한 고찰

2017년 2월

서울대학교 대학원

수리정보학과 디지털포렌식 전공

진 종 민

데이터베이스 은닉행위 식별을 위한 System Artifacts 조사 기법에 대한 고찰

지도교수 천 정 희

이 논문을 이학석사 학위논문으로 제출함
2016년 12월

서울대학교 대학원
수리정보과학과 디지털포렌식 전공
진 종 민

진종민의 석사 학위논문을 인준함
2016년 12월

위 원 장 이 효 원 (인)

부위원장 박 상 준 (인)

위 원 천 정 희 (인)

국문초록

압수수색 영장 집행과정에서 데이터베이스의 조사 또는 수색 시, 디지털 포렌식 수사관은 다양한 상황에 직면하게 된다. 그에 따라 현장 상황을 신속하게 판단하고 적합한 대응을 하여야 한다. 수사관의 현장 상황에 대한 판단 및 대응은 압수수색 결과에 많은 영향을 미친다. 최근 압수수색 현장에서 데이터베이스에 저장된 범죄 혐의와 관련된 데이터를 삭제하거나 데이터가 저장된 제장매체 또는 파일을 은닉한 정황을 흔치 않게 목격하게 된다.

이러한 데이터베이스 은닉행위가 발생하였을 경우 압수수색 현장에서는 법률적, 기술적, 절차적인 관점에서 다각적인 검토가 필요하다. 우선 법률적인 관점에서 혐의사실과 관련 없는 부분을 포함한 전체 데이터 파일의 압수를 위한 판단 근거가 필요하다. 또한 본 건의 범죄사실 이외에 추가적인 별도의 증거인멸 혐의에 대한 증거 확보가 요구된다. 은닉행위를 식별하기 위해서는 데이터베이스와 이와 연관되어 작동하는 시스템에서 생성되는 아티팩츠를 조사 및 수집하여 한다. 하지만 압수수색 당일 현장에서 사용자의 행위 분석을 위하여 시스템에서 생성된 모든 아티팩츠를 조사한다는 것을 현실적으로 불가능하다. 따라서 수많은 아티팩츠 중 사전에 데이터베이스의 은닉행위와 관련된 아티팩츠를 정의하여야 한다. 또한 위 법률적, 기술적인 검토사항을 반영하여 압수수색 전 과정을 효율적으로 진행하기 위한 구체적인 계획 또는 절차가 필요하다.

이에 본 논문에서는 압수수색 현장에서 피 압수자의 데이터 은닉행위를 식별할 수 있는 시스템 아티팩츠의 조사 방법에 대해 고찰하였다. 피 압수자의 은닉행위를 하드디스크 교체, 데이터베이스 교체, 레코드 삭제

의 세 가지 유형으로 구분하여 정의하였다. 각 유형 별로 은닉행위자가 수행하게 되는 수반행위 따라 생성되는 시스템 아티팩츠를 선정하고 이에 대한 조사 방법을 제시하였다. 이러한 과정을 통해서 압수수색 현장에서 은닉행위가 발생할 경우, 행위 유형에 따른 조사와 압수수색 절차를 구분하여 진행함으로써 압수수색 전과정의 효율성을 높였다.

주요어 : 디지털포렌식, 데이터베이스, 은닉행위, 아티팩츠, 조사방법, 절차

학 번 : 2015-26067

목 차

표 목 차

그 림 목 차

Ⅰ. 서론

압수수색 영장 집행과정에서 데이터베이스의 조사 또는 수색 시, 디지털 포렌식 수사관은 다양한 상황에 직면하게 된다. 그에 따라 현장 상황을 신속하게 판단하고 적합한 대응을 하여야 한다. 수사관의 현장 상황에 대한 판단 및 대응은 압수수색 결과에 많은 영향을 미친다. 다시 말해, 디지털 포렌식 수사관은 압수수색 현장에서 범죄와 관련된 데이터의 저장 위치와 형태를 파악하고 범죄 사실과 관련된 데이터를 추출하기 위한 효과적인 방법을 모색하고 결정하여야 한다. 이와 더불어 각 전산시스템에서 생성되는 로그 등의 시스템 아티팩츠을 분석하여 사용자의 행위 분석 및 증거인멸 유무 등을 조사하여야 한다.

최근 압수수색 현장에서 데이터베이스에 저장된 범죄 혐의와 관련된 데이터를 삭제하거나 데이터가 저장된 제장매체 또는 파일을 은닉한 정황을 흔치않게 목격하게 된다. 이러한 데이터베이스 은닉행위가 발생하였을 경우 압수수색 현장에서는 법률적, 기술적, 절차적인 관점에서 다각적인 검토가 필요하다.

데이터베이스의 복구 작업을 위해서는 전체 데이터베이스의 압수가 요구된다. 전체 데이터베이스에는 혐의사실과 관련 없는 부분이 포함되어 있기 때문에 이러한 전체 데이터의 압수에 대한 판단 근거가 필요하다. 또한 본 범죄사실 이외에 추가적인 증거인멸 혐의사실에 대한 증거의 수집이 요구된다. 현재까지 다양한 데이터베이스 별 삭제된 레코드의 복구 기법에 대한 활발한 연구가 진행되었다. 이러한 연구에서는 데이터베이스의 작업내역을 기록한 트랜잭션 로그의 분석[1][2] 또는 데이터파일 구조

1) Paul M. Wright, Oracle Database Forensics using LogMiner, SANS London June 2004 Conference(2005)

의 분석[3]을 통하여 삭제된 레코드의 복구를 수행한다. 하지만 이러한 복구 수행을 위해서는 혐의사실과 무관한 정보가 포함된 전체 데이터베이스의 데이터 파일 또는 트랜잭션 로그 파일의 압수가 선행되어야 한다. 하지만 다음의 우리나라의 현행 “형사소송법 제 106조”와 “압수수색 영장의 압수 대상 및 방법의 제한”에 따라 압수수색 현장에서 데이터의 전체 압수는 예외적인 경우에 한하여 가능하다. 즉, 수사관은 압수수색 현장에서 전체 데이터를 압수하는 경우 “압수의 목적을 달성하기 현저히 곤란하다고 인정되는 때”라고 판단한 근거를 소명할 수 있어야 한다.

[형사소송법] 제106조(압수)

(중략)

③ 법원은 압수의 목적물이 컴퓨터용 디스크, 그 밖에 이와 비슷한 정보저장매체(이하 이 항에서 "정보저장매체 등"이라 한다)인 경우에는 **기억된 정보의 범위를 정하여 출력하거나 복제하여 제출**받아야 한다. **다만, 범위를 정하여 출력 또는 복제하는 방법이 불가능하거나 압수의 목적을 달성하기에 현저히 곤란하다고 인정되는 때**에는 **정보저장매체 등을 압수할 수 있다.**

[압수수색 영장 별지] 압수 대상 및 방법의 제한

2. 컴퓨터용 디스크 등 정보저장매체에 저장된 전자정보에 대한 압수·수색·검증

(중략)

나. 전자정보의 압수

(1) 원칙 : 저장매체의 소재지에서 수색·검증 후 **혐의사실과 관련된 전자정보만을 범위를 정하여** 문서로 출력하거나 수사기관이 휴대한 저장매체에 복제하는 방법으로 압수할 수 있음.

---

2) Paul S. Randal, Understanding Logging and Recovery in SQL Server, microsoft TechNet Magazine(2009. 2.)

3) 박수영, 데이터베이스 내 삭제된 레코드의 복원 방법에 관한 연구, 고려대학교(2013)

(2) **저장매체 자체를 반출하거나 하드카피·이미징 등 형태로 반출할 수 있는 경우**

(가) **저장매체 소재지에서 하드카피·이미징 등 형태(이하 "복제본"이라 함)로 반출하는 경우**

- 혐의사실과 관련된 전자정보의 범위를 정하여 출력·복제하는 **위 (1)항 기재의 원칙적 압수 방법이 불가능하거나, 압수 목적을 달성하기에 현저히 곤란한 경우에 한하여,** 저장매체에 들어 있는 전자파일 전부를 하드카피·이미징하여 그 복제본을 외부로 반출할 수 있음.

(나) 저장매체의 원본 반출이 허용되는 경우

1) 위 (가)항에 따라 집행현장에서 저장매체의 복제본 획득이 불가능하거나 현저히 곤란한 때에 한하여, 피압수자 등의 참여 하에 저장매체 원본을 봉인하여 저장매체의 소재지외의 장소로 반출할 수 있음.

사용자의 행위 분석은 시스템이나 응용프로그램이 자동으로 생성한 아티팩츠의 조사를 통해 가능하다. 하지만 제한된 시간 내에 데이터베이스 전담 수사관이 범죄혐의와 관련된 데이터를 탐색 및 추출 작업을 수행하는 것 이외에, 사용자의 행위 분석을 위하여 모든 시스템 아티팩츠를 조사한다는 것은 사실상 불가능하다. 통상적으로 압수수색 시 하나의 회사에는 한명의 데이터베이스 전담 수사관이 배정된다. 데이터베이스 전담 수사관은 전자결재, 이메일, 회계 시스템 및 업무 유형에 따라 독자적으로 개발한 시스템 등 다양한 종류의 시스템과 그에 저장된 방대한 양의 데이터를 조사하여야 한다. 또한 이러한 장비에 사용되는 운영체제, 응용프로그램의 종류는 수십 가지이고, 각 프로그램에서 생성되는 아티팩츠의 종류와 형태도 제 각각이기 때문이다. 이와 관련하여 사이버침해사고가 발생하였을 경우 디지털 증거의 식별·수집에 관해 많은 연구[4]가 진

행되었다. 하지만 데이터 은닉행위는 행위자가 시스템에 접근 권한을 소유하고 있으므로, 무단으로 시스템에 접근한 침해행위와는 구별되며, 조사하여야 할 아티팩츠에도 차이가 있다. 따라서 수많은 아티팩츠 중에서 데이터베이스의 은닉행위의 식별을 위해서 조사하여할 대상을 선정하여야 한다.

또한 위와 같은 검토사항을 반영하여 압수수색 전 과정을 효율적으로 진행하기 위해서는 구체적인 계획 또는 절차가 마련되어야 한다. 이와 관련하여 데이터베이스의 구조 분석을 통한 일반화된 분석 프로세스[5]와 각 제조사별 DBMS의 디지털 포렌식 조사 절차 및 기법[6]에 대해서는 많은 연구가 진행되었다. 이러한 연구 결과를 토대로 “데이터베이스 은닉행위”라는 특정 상황에서의 조사대상 및 압수수색 진행 절차에 대한 연구가 필요하다.

이에 본 논문에서는 압수수색 현장에서 피 압수자가 범죄 혐의와 관련된 데이터베이스에 저장된 데이터를 은닉하는 상황이 발생하였을 경우 시스템 아티팩츠의 조사를 통해서 그러한 은닉행위를 식별하는 방법에 대해 고찰하였다. 피 압수자의 은닉행위를 하드디스크 교체, 데이터베이스 교체, 레코드 삭제의 세 가지 유형으로 구분하여 정의하였다. 그리고 각 유형 별로 은닉행위자가 수행하게 되는 수반행위를 선정하고 그에 따라 생성되는 시스템 아티팩츠에 대한 조사 방법을 제시하였다.

---

4) 신경준, 사이버 침해사고 유형별 디지털포렌식 증거의 식별 및 수집에 관한 연구, 고려대학교(2007)

5) 김성혜 외 3명, 분석 목적별 분류기반의 데이터베이스 포렌식 모델, 고려대학교(2008)

6) Harmeet Kaur Khanuja and D.S.Adane, A FRAMEWORK FOR DATABASE FORENSIC ANALYSIS, Computer Science & Engineering: An International Journal (CSEIJ), Vol.2,(2012)

본문의 구성은 다음과 같다. 2장에서 수많은 시스템 중에서 데이터은닉행위와 관련된 것을 선별하고, 선별된 시스템에서 사용되는 프로그램 중에서 조사대상을 선정한다. 3장에서는 조사대상으로 선정된 프로그램의 각 아티팩츠의 공통적인 특성에 따라 분류하고 각 특성을 파악한다. 4장에서는 데이터베이스의 은닉행위를 유형화하고 그에 따른 조사대상 및 조사방법과 조사 시 고려하여할 사항에 대해 기술한다. 5장에서는 행위 유형별 시스템 아티팩츠의 조사를 통한 전반적인 압수수색 절차의 진행 방법과 조사 대상 아티팩츠를 일괄하여 추출할 수 있는 방법에 대해 고려해 본다. 6장에서는 결론 및 향후 연구방향을 제시한다.

Ⅱ. 시스템 아티팩츠의 조사 범위 선정

전산시스템의 데이터의 대부분은 데이터베이스에 저장된다. 압수수색 현장에서 이러한 데이터베이스에 저장된 데이터를 누군가 고의적으로 삭제 또는 은닉하는 상황이 발생하였다고 가정하자. 이러한 데이터베이스의 은닉행위(이하 은닉행위)에 대해서 디지털포렌식 수사관은 은닉행위를 수행한 자(이하 은닉행위자, 행위주체), 은닉한 시점(범죄일시), 은닉행위가 발생한 시스템 또는 접속한 PC(범죄장소), 은닉행위의 대상(객체), 수행한 은닉행위(범죄행위) 등에 대하여 규명하여야 한다.

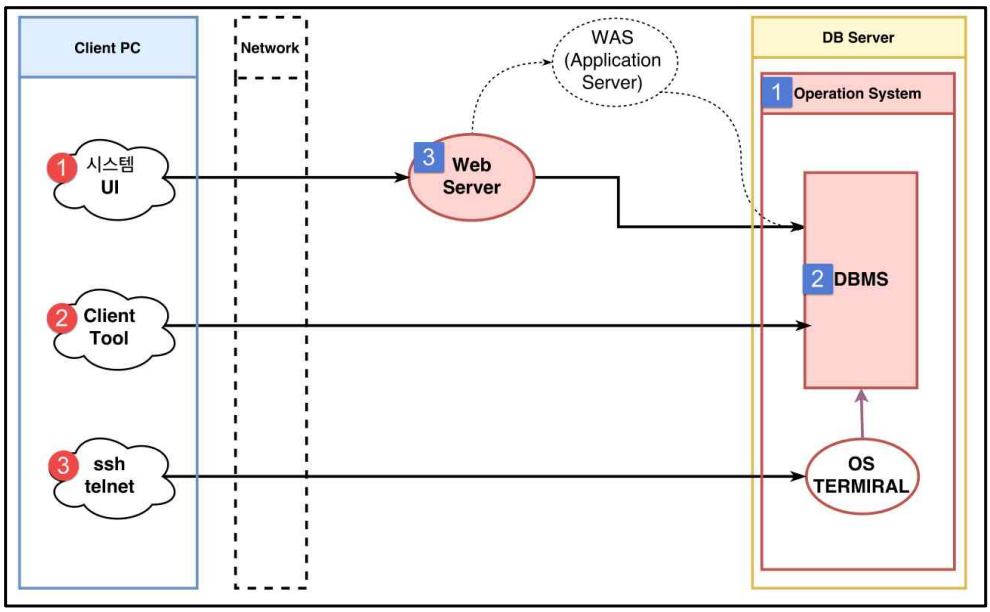
이러한 은닉행위의 규명은 각각의 전산시스템에서 생성되는 시스템 아티팩츠(System Arifacts, 이하 아티팩츠)의 분석을 통해서 가능하다. 디지털포렌식 분야에서 아티팩츠(Artifacts)의 의미는 사용자가 컴퓨터를 사용하게 되면 운영체제나 응용프로그램에서 자동으로 생성한 데이터를 말한다. 이렇게 생성된 아티팩츠를 분석함으로서 사용자가 컴퓨터를 통하여 수행한 행위를 분석할 수 있다.

은닉행위 식별을 위한 시스템 아티팩츠를 조사(이하 조사)함에 있어 다음에서 기술할 사항을 고려하여 조사 대상의 범위를 정하였다.

1. 데이터베이스의 접근 유형에 따른 조사대상 선별

우선 은닉행위자가 데이터베이스의 은닉행위를 하기 위해서 원거리 또는 원격지에 소재하는 데이터베이스 서버에 어떻게 접근할 수 있는지 살펴보자. 은닉행위자는 [그림 2-1]과 같이 세 가지 데이터베이스의 접속 유형에 따라 데이터베이스에 접근할 수 있다. 각 접근 유형에 따라 연관

동작하는 네트워크 장비, 웹 서버, 데이터베이스 서버 등 시스템에서는 각각의 아티팩츠가 생성된다.



[그림 2-1] 은닉행위자의 데이터베이스 접근 유형

사용자가 데이터베이스의 데이터를 변경하기 위해서는 서버 또는 하드디스크 자체를 물리적으로 은닉하는 방법을 제외한다면 데이터베이스 또는 운영체제에 접근하여 데이터의 조작을 위한 명령어를 수행하여야 한다. 원거리의 클라이언트 PC상에서 위 [그림 2-1]의 같이 ① 기업 내의 전산시스템에서 제공하는 시스템의 사용자 인터페이스(User Interface)를 이용하거나, ② 데이터베이스 접속을 위한 클라이언트 도구를 이용하거나, ③ 운영체제의 터미널에 접속(SSH 등의 프로토콜 이용)하여 데이터베이스에 접근 후 데이터의 검색 및 변경 등을 작업을 수행할 수 있다.

이 경우 은닉행위자가 수행한 명령은 기본적으로 사내 또는 사외의 라

우터, 스위치 등의 네트워크 장비를 통하여 전송되며, 경우에 따라서는 설치된 방화벽, 침입탐지장비와 같은 보안장비를 거치게 된다. 하지만 본 논문에서 다루고 있는 은닉행위는 압수수색을 대비하여 회사 내 전산시스템의 접근 권한을 가진 사용자 또는 관리자가 수행하는 작업을 대상하고 있다. 따라서 네트워크 장비와 보안장비에 기록되는 로그 등은 권한이 없는 사용자가 권한을 탈취하기 위한 접근이나 비정상적인 트래픽을 송신과 같은 침해사고 시에 조사대상이 된다. 따라서 은닉행위와는 직접 연관성이 없기 때문에 조사 대상에서 제외하여야 한다.

위와 같이 권한 있는 은닉행위자가 시스템에 위 세 가지 유형으로 데이터베이스 시스템에 접속 후 데이터를 조회 및 변경한 경우, 은닉행위의 흔적을 추적하기 위해서는 데이터베이스 시스템에서 발생하는 아티팩츠와 더불어 해당 데이터베이스 시스템과 연관하여 작동하는 다른 시스템들의 아티팩츠을 동시에 조사하여야 한다.

다시 말해, ①의 경로에서 사용자가 웹 서버의 서비스를 이용하도록 구축된 시스템의 사용자 인터페이스를 이용하여 데이터베이스 시스템에 접속 후 데이터 삭제 등의 은닉행위를 하였다면 해당 [3]의 웹서버 시스템의 접속로그와 [2]의 데이터베이스 관리시스템의 메타데이터 및 각종 로그에 사용자의 흔적이 기록된다. 또한 위 두 개의 시스템에 기반이 되는 [1]의 운영체제 시스템 또한 은닉행위와 관련된 아티팩츠가 생성된다. ②의 경로를 통하여 데이터베이스에 직접 접속하여 은닉행위를 하였을 경우 [2]의 데이터베이스 시스템의 아티팩츠는 물론 기반이 되는 [1]의 운영체제에도 관련 정보를 찾을 수 있다. ③의 경로과 같이 [1]의 운영체제의 터미널에 접속하여 데이터베이스에 접속하거나 관련된 명령어를 수행한다면 데이터베이스 시스템과 운영체제에서 은닉행위에 대한 아티팩츠를 수집할 수 있다.

위 세 가지 경우를 종합하여 보면 피 압수자의 데이터베이스의 은닉행위와 관련된 시스템 아티팩츠가 생성 및 기록되는 시스템은 데이터베이스 시스템과 웹 서버 시스템 그리고 이러한 응용프로그램이 설치된 운영체제이다. 즉 운영체제, DBMS, 웹 서버 시스템이 데이터 은닉행위와 관련된 조사의 대상이 된다.

2. 시장 점유율에 따른 조사대상 선정

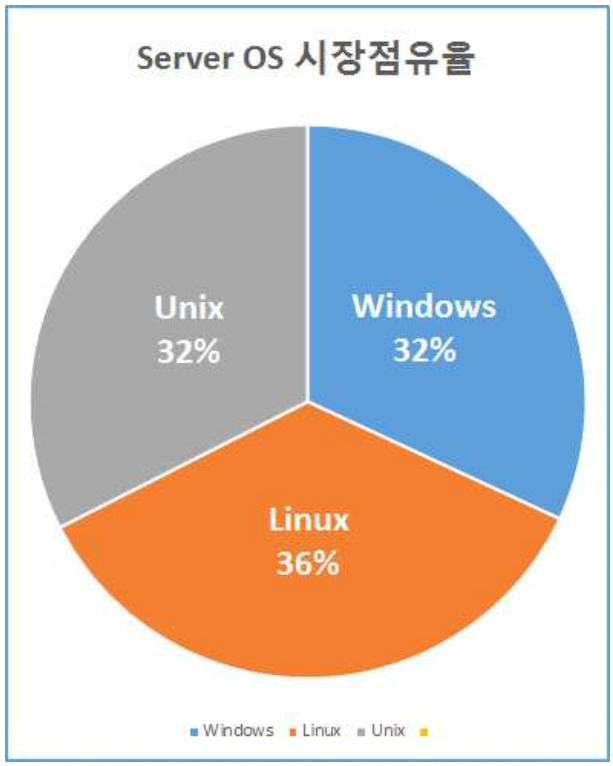
하지만 1절과 같이 압수수색 현장 내의 네트워크 상의 모든 장비 중에서 세 가지 시스템(운영체제, DBMS, 웹서버)를 특정하더라도 또 다른 문제가 있다. 세 가지 시스템에서 각각 사용하는 프로그램의 종류의 수는 매우 많다는 것이다. 또한 시스템 아티팩츠는 각 제조사의 특성 및 시스템 설정에 따라 기록되는 정보의 종류 및 보관방법들에 차이가 있다. 따라서 압수수색 현장 내의 “모든 아티팩츠”에 대한 “사전 지식을 모두 습득”한다는 것은 사실 상 불가능하다.

이에 시장점유율[7]이 높을수록 현장에서 마주칠 기회가 많은 시스템이라 가정하고, 시장 점유율 20% 이상인 시스템을 본 논문의 연구대상으로 선정하였다.

다음 [그림 2-2]의 시장 점유율을 살펴보면 우선 첫 번째 그림에서 서버에 설치되는 운영체제의 경우 2015년을 기준으로 리눅스(Linux) 시스템 30%, 윈도우(Windows Server) 시스템 32%, 유닉스(Unix) 시스템 32%로 메인프레임을 제외한다면 3가지의 운영체제가 비슷한 비율로 점유하고 있다. 유닉스 시스템은 리눅스 시스템과 그 구조가 매우 유사하

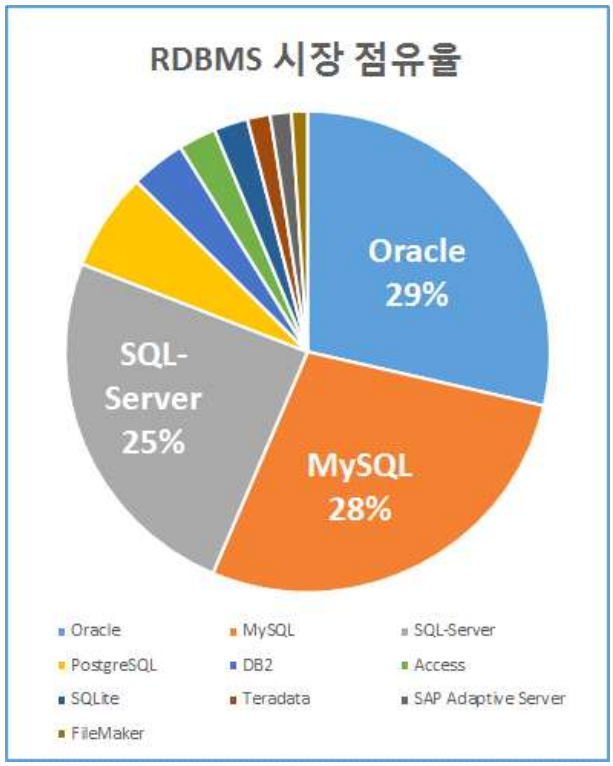
7) WIKIPEDIA, Usage_share_of_operating_systems, https://en.wikipedia.org/wiki/Usage_share_of_operating_systems

기 때문에 논문의 연구대상에서 제외하였다.



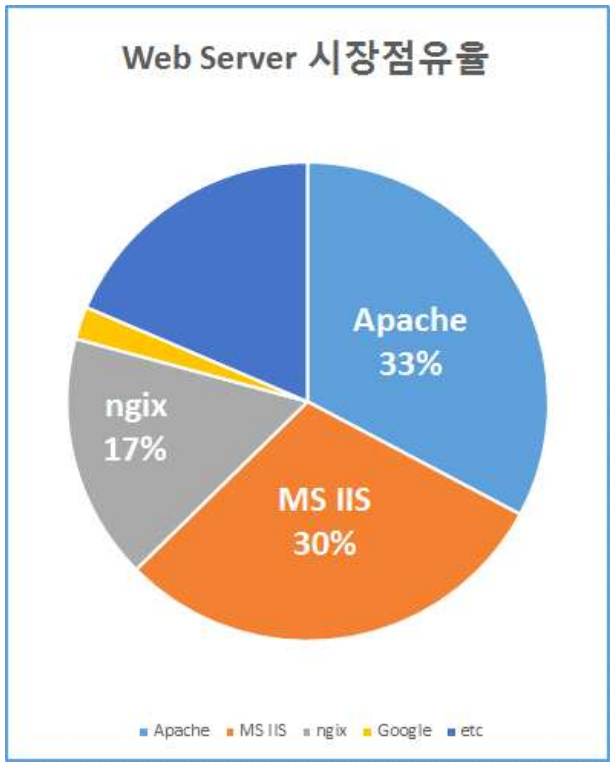
[그림 2-2] 2015 Sever OS 시장점유율(W3Tech)

다음 [그림 2-3]의 데이터베이스 관리시스템(DBMS)의 시장 점유율을 살펴보면 2016년을 기준으로 Oracle 29%, MySQL 28%, SQLSever 25%로 나머지 19%를 제외하면 3개의 제품이 비슷한 비율로 점유하고 있다. 일반적으로 이러한 통계적인 요소보다는 회사의 규모에 따라 사용하는 DBMS 소프트웨어의 차이가 발생한다.



[그림 2-3] 2016 DBMS 시장점유율(DB-ENGINES)

마지막으로 [그림 2-4]의 웹 서버 시장점유율를 살펴보면 2016년 기준으로 20%가 넘는 제품은 Apache 33%, MS IIS 30% 두 종류 뿐이다. 따라서, 대부분의 웹 서버는 이 두 제품을 사용한다고 보아야 한다.



[그림 2-4] 2016 웹 서버 시장점유율(Netcraft)

이러한 통계를 바탕으로 다음 [표 2-1]과 데이터베이스 은닉행위와 관련된 운영체제(리눅스, 윈도우), 데이터베이스 관리 시스템(Oracle, MS SQL-Sever, MySQL), 웹 서버(Apache, MS IIS)를 본 논문의 연구 대상으로 선정하였다.

운영체제	DBMS	웹서버
○ MS Windows Server ○ Linux	○ Oracle ○ MS SQL-Server ○ MySQL	○ Apache ○ MS IIS

[표 2-1] 연구 대상 응용프로그램 선정

Ⅲ. 시스템 아티팩츠의 분류 및 특징

시스템 아티팩츠는 프로그램의 성능 및 보안 등의 구동 목적에 따라 다양한 형태로 생성된다. 또한 프로그램의 제조사는 타 프로그램에서 사용하는 아티팩츠와 유사한 기능을 보유하더라도 그 명칭은 독자적으로 명명하고 있다.

따라서, 이 장에서는 본 논문과 관련된 시스템 아티팩츠를 공통된 특성에 따라 분류 후 주요한 기능을 살펴본다. 또한 세분하여 각 프로그램별로 독특한 특성을 파악한다.

1. 운영체제(Operating System)

가. 시스템 로그

로그란 감사 설정된 시스템의 모든 기록을 담고 있는 데이터이다. 이러한 데이터는 성능, 오류, 경고 및 운영 정보 등의 중요 정보가 기록되며, 특별한 형태의 기준에 따라 숫자와 기호 등으로 이루어져 있다. 이러한 로그를 분석하여 일반적으로 시스템의 장애 원인 분석 관리 및 침입 감지, 추적 등의 필요한 유용한 정보를 알아 낼 수 있다.

대다수의 운영제체는 로깅 능력이 있으며, 발생되는 이벤트를 수집하여 보관한다. 시스템 로그의 조사 시 유의할 점은 다음과 같다.

○ 시스템이 저장하는 로그 파일은 서버의 운영체제에 따라 그 유형의 차이가 있고 그 종류가 다양하다.

○ 설정 값에 따라 로그를 기록하는 위치와 보관 주기 및 용량에 차이

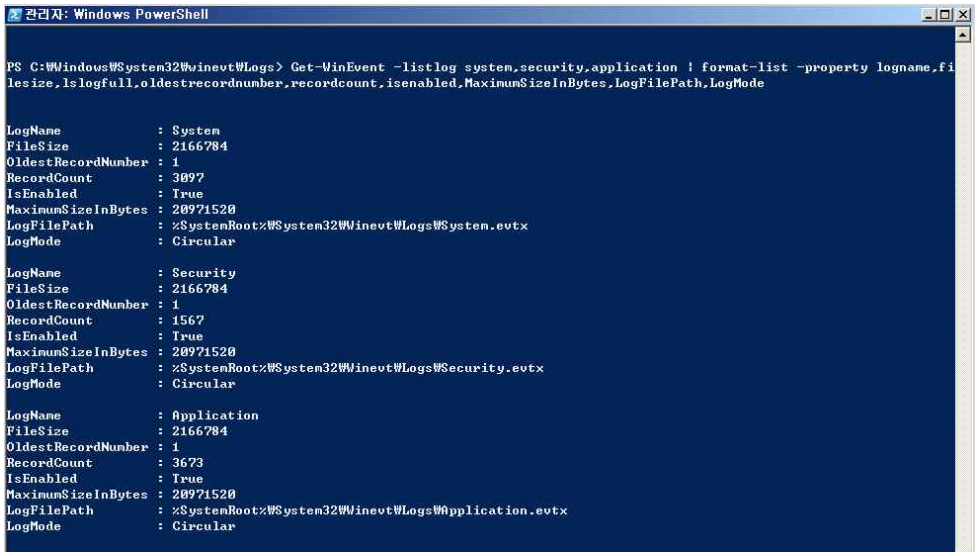
가 있다.

○ 조사시스템의 로컬 디스크에 로그를 저장하지 않고 네트워크로 연결된 원격의 로그 서버나 연동된 데이터베이스 서버에 데이터를 저장하는 경우, 이에 대한 추가 조사가 이루어져야 한다.

(1) 윈도우 이벤트 로그(Windows Event Log)

윈도우 이벤트 로그의 조사를 통해서 사용자 계정의 로그인/로그오프, 원격 데스크톱 계정의 활동, 응용프로그램의 실행/종료, 데이터베이스의 실행/종료와 관련된 이벤트의 정보를 얻을 수 있다.

윈도우 시스템에서는 시스템의 로그를 이벤트 로그 형식으로 관리한다. 이벤트 로그는 시스템에서 발생하는 시스템 로그 및 응용프로그램 로그, 보안 로그와 같은 전역 로그 이외 윈도우 파워셀, 인터넷 익스플로러, 하드웨어 이벤트, 윈도우 구성 요소 이벤트 등 200여개의 이벤트들을 기록한다.



[그림 3-1] Powershell를 통한 이벤트 로그 속성 값 확인

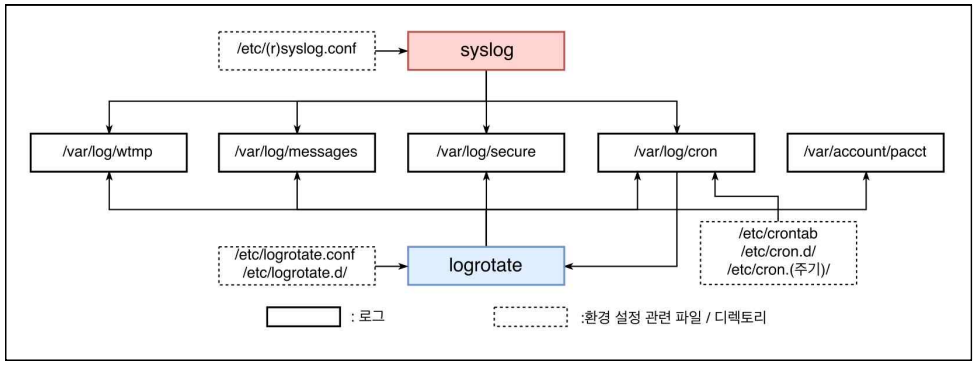
조사 시 [그림 3-1]과 같이 해당 이벤트 로그의 가장 오래된 로그번호(OldestRecordNumber), 최대크기(MaximumSizeInByte), 보관주기(LogMode, default : circular)를 확인하여 이벤트가 덮어쓰기 된 경우에 대한 시간적 고려가 필요하다.

(2) 리눅스 시스템 로그

리눅스 시스템 로그의 조사를 통해 계정별 로그인/로그아웃, 원격접속, 계정별 명령어 사용 내역, crond 작업내역 등의 정보를 확인할 수 있다.

syslog는 커널과 많은 응용프로그램이 메시지를 로깅 할 목적으로 유닉스 기반 시스템에서 사용하는 일반적인 방식이다. 시스템 로그 데몬(syslogd)에 의해서 실행되고 배포판에 따라서 syslog, rsyslog, syslog-NG(syslog New Generation)를 사용한다. 로그파일별로 저장되는 디렉토리의 위치 및 저장형식은 /etc/(r)syslog.conf 파일에서 설정한다.

위와 같은 로그파일들은 [그림 3-2]와 같이 logrotate 데몬에 의해서 순환하여 저장된다.

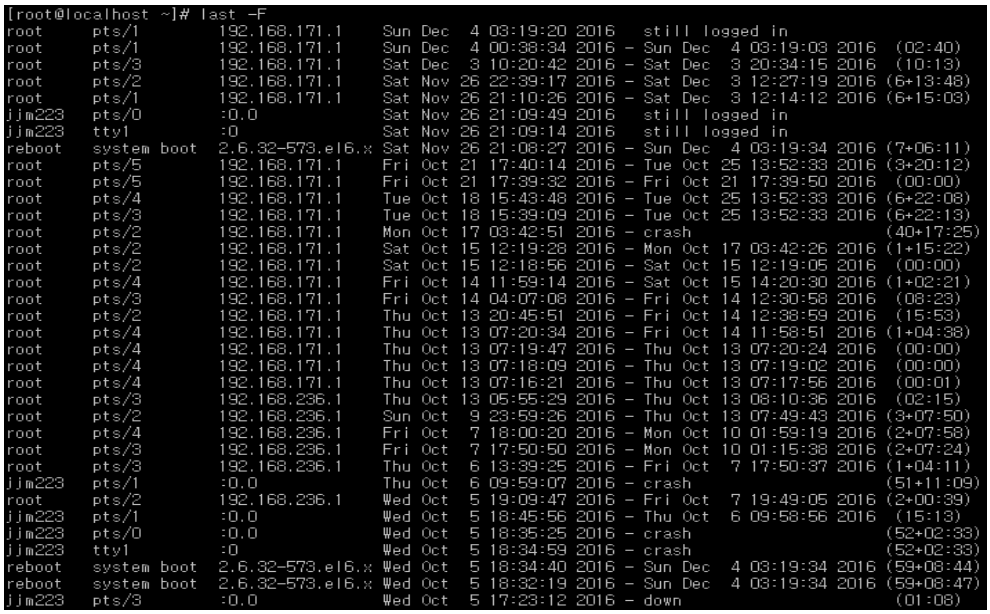


[그림 3-2] 리눅스 시스템 로그 관리

logrotate 데몬은 /etc/cron.daily 디렉토리에 다음과 같이 cron 작업으로 등록되어 있어 매일 주기적으로 실행되며 /etc/logrotate.conf 파일과 /etc/logrotate.d/ 디렉토리 내에 있는 파일들의 설정된 조건에 따라 순환 작업이 발

생한다. 따라서 syslogd, crond, logrotated의 환경설정파일의 조사를 통해 보관되는 로그의 종류 및 보관주기에 대한 고려가 필요하다.

리눅스 시스템 로그를 통합적으로 관리하거나 로그 내용을 확인할 수 있는 도구 등이 존재하지만, 별도의 설치가 요구되므로 [그림 3-3]과 같이 운영체제에서 제공하는 명령어를 사용하는 것이 압수수색 현장에서는 좀 더 유용하다.



[그림 3-3] 리눅스 접속계정 정보 조회

나. 윈도우 레지스트리(Registry)

윈도우 시스템에서는 설치된 소프트웨어 정보, 환경설정, 임시 저장 값, 인스톨된 프로그램 등의 정보를 담고 있으므로 운영체제와 데이터베이스 관리시스템 및 웹 서버 등을 포함한 응용프로그램의 설치에 관한 흔적을 레지스트리로부터 찾을 수 있다.

리눅스 시스템과 달리 윈도우 시스템은 운영에 필요한 환경설정 등의 데이터를 통합하여 레지스트리에서 관리한다. 레지스트리는 윈도우 3.1 이후 버전부터 운영체제와 응용프로그램의 필요한 정보를 저장하기 위해 고안된 계층형 데이터베이스이다. 아래 [표 2-2]는 본 논문과 관련하여 운영체제 및 응용프로그램의 설치일시가 저장된 레지스트리 키이다.

내용	참조키
OS 설치일시	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft \Windows NT\CurrentVersion\InstallDate
응용 프로그램 설치일시	▪ HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion \Uninstall* ▪ HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall*

[표 3-1] 운영체제, 응용프로그램 설치일시 관련 레지스트리 키

압수수색 현장의 활성시스템에서 레지스트리의 조사를 위해서는 윈도우 시스템에서 기본 제공하는 "regedit"를 이용하거나 다음과 같이 powershell을 이용하여 각 항목의 값을 불러올 수 있다.

```
PS C:\> Get-ItemProperty Registry::"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion" | Select-Object productname, productid, installdate | Format-List

ProductName : Windows Server 2008 R2 Datacenter
ProductId   : 00496-001-0001283-84696
InstallDate : 1460354156
```

2. 데이터베이스 관리 시스템(DBMS)

DBMS(DataBase Management System)는 다수의 사용자들이 데이터베이스를 효율적으로 관리하고 접근, 조작하기 위한 관리용 소프트웨어의 집합이다.[8)]

DBMS는 관리의 편의성을 위하여 메타데이터 및 각종 로그를 생성하고 관리하다. 이러한 메타데이터와 로그들의 수집 및 분석을 통하여 사용자의 접근형태 및 행위를 도출할 수 있다.

가. 메타데이터(Metadata, 시스템 테이블)

각 데이터베이스의 메타데이터로부터는 압수수색 현장에서 데이터베이스의 생성과 관련된 정보, 사용자의 로그인 기록 및 작업내역 등의 정보를 추출 및 조사할 수 있다.

메타데이터는 다른 데이터를 설명해 주는 데이터를 말한다. 메타데이터의 역할은 자원을 효율적으로 관리하기 위해서 계정의 권한, 테이블과 컬럼의 데이터 타입, 데이터베이스의 저장구조 등의 정보를 목록화하여 저장하기 위함이다.

DBMS는 종류 및 기능에 따라 메타데이터들로 구성된 별도의 시스템 데이터베이스(또는 테이블)을 운영한다. 이러한 시스템테이블은 다음과 같은 몇 가지 특징을 가진다.

○ 사용자가 테이블을 생성하거나 사용자를 변경하는 등의 작업을 할 때 데이터베이스 서버에 의해 자동으로 갱신된다.

---

8) WIKIPEDIA, 데이터베이스 관리 시스템, https://ko.wikipedia.org/wiki/데이터베이스_관리_시스템

○ 기본테이블은 데이터베이스 서버만 기록할 수 있고, 사용자가 직접 접근 할 수 없다. 이 때문에 데이터베이스는 기본테이블을 사용자가 이해할 수 있는 형태인 뷰나 시스템 함수, 시스템 저장프로시저로 접근 방법을 제공한다.

○ 시스템의 성능과 관련된 동적테이블의 경우 데이터베이스의 종료와 함께 그 정보를 상실하는 경우가 발생할 수 있다.

이러한 시스템테이블은 INFORMATION_SCHEMA라는 규격화된 포맷이 존재하지만 각 DBMS는 제조사 별로 독자적인 형태 및 명칭을 사용한다. 각 제조사별 제공하는 뷰의 형태 및 특징은 다음과 같다.

○ Oracle(Data Dictionary)[9]

- 총 4개의 카테고리로 분류

구분	접두어	설명
Static Dictionary	USER_	해당 사용자가 생성한 객체만 조회 가능
	ALL_	사용자와 사용자에 권한이 부여된 객체만 조회 가능
	DBA_	데이터베이스에서 생성되는 모든 내용 조회 가능
Dynamic Dictionary	v$	▪ 실시간으로 조회하는 시점의 데이터를 메모리 구조로부터 작성된 가상 테이블을 기반으로 조회 ▪ 세션, 파일상태, 작업 진행 상황, Lock, 백업 상태, 메모리 사용 및 할당, 시스템 및 세션 파라미터, SQL 실행, 통계 및 Metric 등의 정보 포함

9) Oracle, Oracle Database 11g: Administration Workshop 1, 188-192

○ SQLServer(System Table)[10)]

- SQLSever 2005 버전 이후 시스템테이블을 직접 조회 불가능

구분	스키마	설명
호환성 뷰 (Compatibility views)		시스템테이블에 익숙한 사용자를 위해 시스템테이블과 비슷한 이름을 가짐
카탈로그 뷰 (Catalog views)	sys.	메타데이터에 접근하기 위한 기본 인터페이스
DMV (Dynamic Management Objects)	sys.dm_	서버의 내부 동작을 모니터할 수 있음 ▪ dm_exe_* : 사용자 코드의 실행과 관련 정보 포함

○ MySQL

- ISO 표준 규격인 INFORMATION_SCHEMA 테이블 구조에 MySQL 고유의 컬럼을 추가함

나. 로그 파일

(1) 트랙잭션 로그(Transaction Log, Redo Log)

데이터베이스는 ACID(Atomity, Consistency, Isolation, Durability)특성을 보장하기 위해서 트랜잭션 단위로 작업을 진행한다. 이러한 트랜잭션을 기록한 로그파일로부터 데이터베이스에서 데이터를 변경한 이력, 실행 DML문장 등의 정보를 얻을 수 있다.

데이터베이스는 트랜잭션이 정상적으로 완료되면 실제 데이터파일에 변경된 데이터를 반영한다. 이와 관련된 트랜잭션 로그는 트랜잭션 역할

10) Kevvie Fowler, SQL SERVER forensic analysis, Addison-Wesley(2009)

을 수행하기 위하여 데이터가 변경되는 경우 그 작업내역 및 변경데이터를 일부 가지고 있다. 트랙잭션 로그는 제조사마다 그 구조는 다르지만 기본적으로 바이너리 파일이므로 제조사에서 제공하는 도구나 상용 도구를 이용하여 그 내용을 읽어 들일 수 있고 일정한 규칙에 따라 순환하며 저장 위치는 시스템에 따라 변경될 수 있다.

트랜잭션로그 조사 시 각 제조사 별 특징 및 유의사항은 다음과 같다.

구분	설명	
Oracle	저장위치	$ORACLE_BASE/oradata/redo##.log
	특징	Archive mode일 경우 별도의 경로에 redo log 파일을 저장
	조사도구	Logminer(Oracle 8i 이후)
SQL Sever	저장위치	C:\Program Files\Microsoft SQL Server\ MSSQL.n\MSSQL\DATA\[DB명].ldf
	특징	복구 모델에 따라 트랙잭션 정보의 저장 형태 차이 있음 ▪ Simple Recovery : 일관성 검사만을 위한 최소한의 트랜잭션 정보만 기록, 마지막 백업시점으로만 복구 가능 ▪ Full Recovery : 백업 이후 모든 트랜잭션 로그 기록, 새로 생성되는 데이터베이스는 기본적으로 Full 모드
	조사도구	▪ SQLSever 제공 : DBCC, fn_dblog(), ▪ 상용도구 : SQL Log Rescue(RedGate), ApexSQL Log SQL Log Analyer(Systools)

구분	설명	
MySQL	저장위치	/var/lib/mysql/mysql-bin.######
	특징	▪ MyISAM Engine은 트랙잭션 기능 없음 ▪ Binary log에 데이터 변경에 대한 모든 문장을 기록
	조사도구	mysqlbinlog

(2) 에러로그(Error Log, Alert Log)

데이터베이스는 운영하면서 발생하는 이벤트를 에러로그에 기록한다. 이러한 에러로그로부터 데이터베이스의 시작, 종료시점에 대한 정보를 확인 할 수 있다.

운영체제가 윈도우 시스템일 경우 이벤트로그와 동시에 기록될 수 있다. 저장위치와 특징은 다음과 같으며 각 시스템 별 환경설정 값에 의해 결정된다.

구분	설명	
Oracle (11g 기준)	저장위치	▪ xml 형태 : $ORACLE_BASE/diag/product_type/product_id /instance_id/alert/log.xml ▪ text 형태 : %ORACLE_BASE/diag/product_type/product_id/instance_id/trace/alert_[SID].log
	특징	각 인스턴스마다 Alert log file(text) 생성

구분	설명	
SQL Sever	저장위치	C:\Program Files\Microsoft SQL Server\MSSQL.n\MSSQL\LOG\
	특징	▪ SQL-Server instance 재시작 시, 새로운 파일 생성 ▪ 7개의 로그 파일 유지 : ERRORLOG : 현재 사용중 ERRORLOG.[1~6] : 이전사용
MySQL	저장위치	/var/log/mysqld.log

다. 트리거(Trigger) 이벤트

트리거는 지정한 table에 insert, update, delete문이 실행되면 자동적으로 수행되는 동작을 말한다. 트리거 이벤트가 존재한다면 데이터 변경 시, 변경 전 데이터를 확보할 수 있고 변경행위가 수행여부에 대한 판단이 가능하다.

아래와 같은 구조로 트리거가 정의되어 있다면 압수수색의 대상이 되는 original(원본) 테이블에서 내용을 삭제할 경우 자동적으로 backup(백업) 테이블에 삭제되기 이전의 데이터를 저장한다면 B테이블에서 삭제되기 이전의 데이터를 확보할 수 있고, 이러한 삭제행위가 수행되었다는 것을 판단할 수 있다.

```
(SQLServer 기준)
CREATE TRIGGER test
ON OriginTable
AFTER
DELETE
AS
  INSERT INTO BackupTable
     SELECT userID, name, addr, height
     FROM deleted;
```

또한 이러한 트리거가 실제 작동에 관한 활성 상태의 확인이 필요하다.

3. 웹 서버(Web Server)

웹 서버의 접속로그(Access Log)는 웹 서버와 브라우저간의 주고 받은 정보들을 기록한 것이다. 일반적으로 접속로그를 웹 로그와 동일한 개념으로 사용하며 다음과 같은 포맷의 종류가 있다.

종류	설명
NCSA CLF (Comon Log Format)	Apache의 기본포멧
NCSA ELF (Extented Log Format)	CLF에 Reference, User-Agent 정보 추가
W3C ELF (Exntented Log Format)	▪ 주로 IIS에서 사용 ▪ Reference, User-Agent를 포함하여, Cookie, 사용자 정보 추가

예를 들어 아파치에서 사용하는 NCSA ELF 포맷에서는 다음과 같이 클라이언트 IP주소, 요청한 처리를 웹 서버가 마친 시간, 클라이언트가 요청한 소스 등에 대한 정보를 얻을 수 있다.

로그 항목	설명
192.168.171.1	클라이언트 IP주소
-	
-	
[27/Nov/2016:00:02:25 +0900]	요청한 처리를 마친 시간
"GET /Mgn/car/regist_ok.html?mode=multi_delete&duid=62294;	클라이언트가 요청한 첫 라인
200	상태코드
181	전송된 바이트
"http://192.168.171.129/list.php"	참조 요청 헤더
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)"	클라이언트 브라우저 식별정보

각 웹 서버 별 로그파일 저장 경로는 다음과 같다.

구분	설명
Apache[11]	/var/log/httpd/access_log-########
IIS[12]	%SystemDrive%\inetpub\logs\LogFiles\W3SVC#[사이트 ID]\

이러한 웹서버의 접속로그는 텍스트 형태로 클라이언트와 서버간의 요청하고 전달한 소스에 대한 간략한 정보만을 저장하므로 해당 웹 소스에 대한 분석이 같이 이루어져야 활용 가능하다.

11) Apache, 로그파일, https://httpd.apache.org/docs/2.2/ko/logs.html
12) MS TechNet, IIS에서 로깅 구성, https://technet.microsoft.com/ko-kr/library/hh831775(v=ws.11).aspx

Ⅳ. 은닉행위 유형 별 조사방법에 대한 고찰

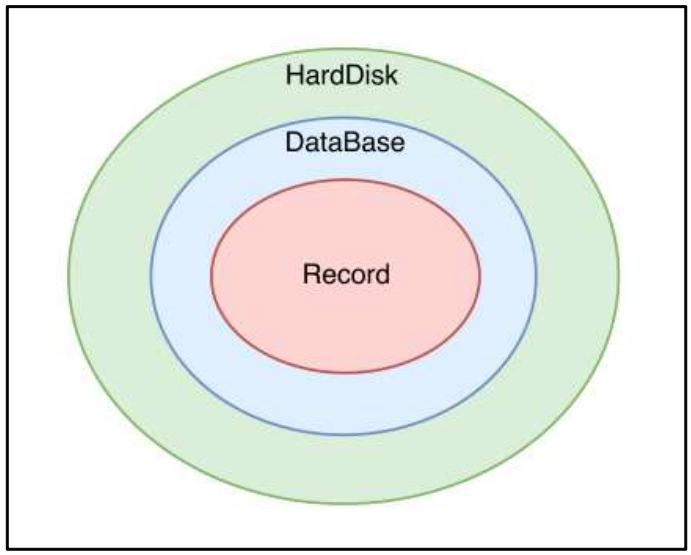
시스템 아티팩츠에 대한 조사 및 분석을 하는 경우 여러 가지 문제에 직면한다. 첫째, 데이터베이스에 저장된 데이터를 은닉하는 방법이 다양하다. 둘째, 하나의 프로그램에서도 생성되는 아티팩츠의 종류가 방대하다. 셋째, 그러한 아티팩츠를 조사 또는 확인할 수 있는 방법이 다양하다. 마지막으로 조사된 시스템 아티팩츠 만으로는 피 압수자의 은닉행위를 명확하게 판단할 수 없는 경우가 있다.

따라서 본 장에서는 데이터베이스의 은닉행위를 세 가지 유형으로 분류하고 행위 유형 별로 수행하는 게 되는 행위(수반 행위) 들을 특정한다. 그 후 수반행위에 따라 생성되는 아티팩츠를 각 프로그램 별로 분류한다. 그리고 그에 따른 효과적인 조사 방법과 고려하여야 하는 사항에 대해 고찰한다.

1. 은닉행위 유형 별 수반행위 선정

가. 은닉행위 유형화의 필요성

물리적 구조를 기반으로 압수수색 현장에서 발생하는 은닉행위는 하드디스크의 교체, 데이터베이스의 교체, 레코드의 삭제로 구분하여 유형화할 수 있다. 데이터베이스에 저장되는 데이터는 [그림 3-1]과 같은 물리적 구조로 포함관계[하드디스크 ⊃ 데이터베이스(파일) ⊃ 레코드]에 있다. 하드디스크 내에 데이터베이스 파일이 저장되며, 데이터베이스 파일 안에 각각의 레코드들이 저장된다.



[그림 4-1] 데이터의 물리적 저장구조

이러한 데이터의 포함관계 때문에 은닉행위 유형에 따른 아티팩츠의 조사대상에 차이가 발생한다. 또한 유형화한 은닉행위에 따라 현장에서 수행해야 할 절차를 구분하여 진행할 수 있다.

나. 은닉행위 유형 별 수반행위 특정

시스템 아티팩츠는 “하드디스크를 교체하였다”, “데이터베이스는 교체하였다”, “레코드를 삭제하였다”는 직접적인 내용을 저장하고 있지는 않다. 즉, 위 사실을 증명하기 위해서는 다음 [표 4-1]과 같은 피 압수사자의 대한 구체적 행위에 대해 조사하여야 한다. 다시 말해, 수반행위에 따라 생성되는 아티팩츠의 수집 및 분석을 통하여 은닉행위자가 어떠한 방법으로 데이터를 은닉하였는지 식별할 수 있다.

데이터베이스 은닉행위 유형		수반 행위
1 하드디스크 교체	→	- 교체한 하드디스크 포맷 - 운영체제 설치 - 응용프로그램 설치
2 데이터베이스 교체	→	- 데이터베이스 On / Offine - 데이터베이스 백업 수행 - 새로운 데이터베이스 생성 - 백업한 데이터베이스 복원
3 레코드 삭제	→	- 데이터 삭제 행위를 위한 DBMS 접속 - DML 명령 수행

[표 4-1] 은닉행위 유형 별 수반행위

위 표를 살펴보면 1 하드디스크의 교체 후 해당 서버를 재사용하기 위해서는 새롭게 설치한 하드디스크를 포맷하고 서버 운영을 위한 운영체제 및 응용프로그램을 설치하여야 한다. 2 데이터베이스의 교체를 통하여 범죄 혐의와 무관한 데이터만을 저장하는 과정에서는 새로운 데이터베이스를 생성하여야 하고, 기업의 주요 핵심 정보를 단순 파기할 수는 없기 때문에 기존의 데이터에 대한 백업 작업이 수행될 수 있다. 이러한 백업 과정에서는 데이터베이스 시스템을 종료 및 재시작하거나 혐의사실과 관련된 데이터베이스만을 오프라인하는 절차가 수반될 수 있다. 3 데이터베이스에서 범죄 혐의사실과 관련된 데이터만을 선별적으로 삭제하기 위해서는 데이터베이스에 접속하여야 하고 이러한 삭제명령을 수행하여야 한다.

다. 고려사항

하지만 이렇게 정의한 수반행위에 대한 아티팩츠가 발견되었다고 해서 반드시 데이터베이스의 은닉을 위한 행동으로 간주할 수는 없다. 회사의 업무 흐름상 이와 같은 행위를 하는 경우는 많기 때문에 이에 대한 보충적 해석이 필요하다.

2. 하드디스크 교체

서론에서 기술한 바와 같이 압수수색 현장에서 그룹웨어로 사용 중인 서버의 하드디스크 전체를 교체하는 사례가 발생하였다. 피 압수자는 이러한 교체 행위를 통하여 회사의 이메일, 전자결재 데이터 전체를 은닉하고자 하였다.

이처럼 은닉행위자 범죄혐의와 관련된 데이터를 은닉하기 위하여, 해당 데이터가 저장된 서버의 하드디스크 전체를 교체하는 경우가 있다. 이러한 경우 새로 장착한 하드디스크에는 운영목적에 맞는 운영체제 및 응용프로그램을 설치하여야 한다. 따라서 하드디스크의 포맷일시 등의 조사를 통하여 교체한 시기 등을 추정할 수 있다. 이 과정에 조사하여야 할 대상과 고려사항은 다음과 같다.

가. 조사 대상

다음 [표 4-2]에서 각 수반행위에 따라 생성되는 아티팩츠를 각 항목에 따라 분류하였다. 압수대상 서버의 사양에 따라 각 항목의 조사를 통하여 하드디스크의 교체 또는 시스템 재설치 일시 등을 추정할 수 있다.

번호	수반 행위	조사대상			
		대분류	중분류	소분류	상세내용
1.1	디스크 포맷 일시	OS	Windows	file system	$MFT의 생성시간
			Linux		Superblock(offset 264-267)[13]의 Filesystem created 값
1.2	운영체제 설치일시	OS	Windows	Registry	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\InstallDate
			Linux	file system	운영체제 설치 후 수정되지 않은 파일의 생성시간
1.3	응용 프로그램 설치일시	OS	Windows	EventLog	Applicaton log (Event ID : 1033)
				Registry	HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall* HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall*
			Linux	System Log	messages Log(설치관련 정보)
				Packaging System	(Red Hat 계열) RPM
					(Red Hat 계열) YUM
					(Debian 계열) dpkg.log

[표 4-2] 하드디스크 교체와 관련된 시스템 아티팩츠

13) WikiKernel, Ext4 Disk Layout, https://ext4.wiki.kernel.org/index.php/Ext4_Disk_Layout#The_Super_Block

각 항목을 살펴보면 운영체제의 파일시스템에서 하드디스크의 포맷 일시에 대한 정보, 윈도우 시스템의 레지스트리에서 운영체제의 설치일시 관련 정보를 확인 할 수 있다. 리눅스 시스템의 경우 최초 운영체제 설치 후 변경되지 않는 파일의 생성시간을 이용하여 운영체제 설치 일시의 확인이 가능하다. 또한 각 운영체제의 시스템 로그 등에서 응용프로그램의 설치와 관련된 정보의 획득이 가능하다.

만약 조사 대상 서버가 리눅스 운영체제를 사용한다면 시스템 아티팩츠 조사를 통하여 다음과 같이 2016. 4. 12. 20:41:38에 하드디스크를 포맷하였고, 2016. 4. 13. 15:19:02에 MySQL, PHP, Apache 프로그램을 설치한 것을 확인할 수 있다. 즉, 하드디스크의 교체 또는 재사용은 2016. 4. 12. 20:41:38 이전에 수행되었다고 추정할 수 있다.

○ 하드디스크 포맷일시(리눅스 시스템, Ext4)

```
[root@localhost /]# mount | awk '/dev.*ext/ {print $1}' | xargs -I DEVICE sh -c '{ mount | grep DEVICE | awk "{ print \"-- device \" \$1 \" mounted on \" \$3 }"; sudo dumpe2fs -h DEVICE | grep "Filesystem created"; }'
-- device /dev/mapper/vg_livecd-lv_root mounted on /
dumpe2fs 1.41.12 (17-May-2010)
Filesystem created:       Sat Aug  8 01:21:50 2015
-- device /dev/sda1 mounted on /boot
dumpe2fs 1.41.12 (17-May-2010)
Filesystem created:       Tue Apr 12 20:41:38 2016
-- device /dev/mapper/vg_livecd-lv_home mounted on /home
dumpe2fs 1.41.12 (17-May-2010)
Filesystem created:       Tue Apr 12 20:41:45 2016
```

○ 응용프로그램 설치일시(리눅스 시스템, yum)

```
[root@localhost ~]# yum history info 2
Loaded plugins: fastestmirror, refresh-packagekit, security
Transaction ID : 2
Begin time     : Wed Apr  13 15:19:02 2016
Begin rpmdb    : 817:eaf4f77dbe1a49a754dafb2ab4cef970d377e6cf
End time       :          15:19:10 2016 (8 seconds)
End rpmdb      : 832:fd42abe28bb582de9c96b31c08140588429c0966
User           : jjm223 <jjm223>
Return-Code    : Success
Command Line   : install httpd mysql mysql-server php php-
mysql
(중략)
Packages Altered:
Dep-Install apr-1.3.9-5.el6_2.x86_64                 @base
Dep-Install apr-util-1.3.9-3.el6_0.1.x86_64          @base
Dep-Install apr-util-ldap-1.3.9-3.el6_0.1.x86_64     @base
(중략)
Install     php-5.3.3-48.el6_8.x86_64                @updates
Dep-Install php-cli-5.3.3-48.el6_8.x86_64            @updates
Dep-Install php-common-5.3.3-48.el6_8.x86_64         @updates
Install     php-mysql-5.3.3-48.el6_8.x86_64          @updates
Dep-Install php-pdo-5.3.3-48.el6_8.x86_64            @updates
history info
```

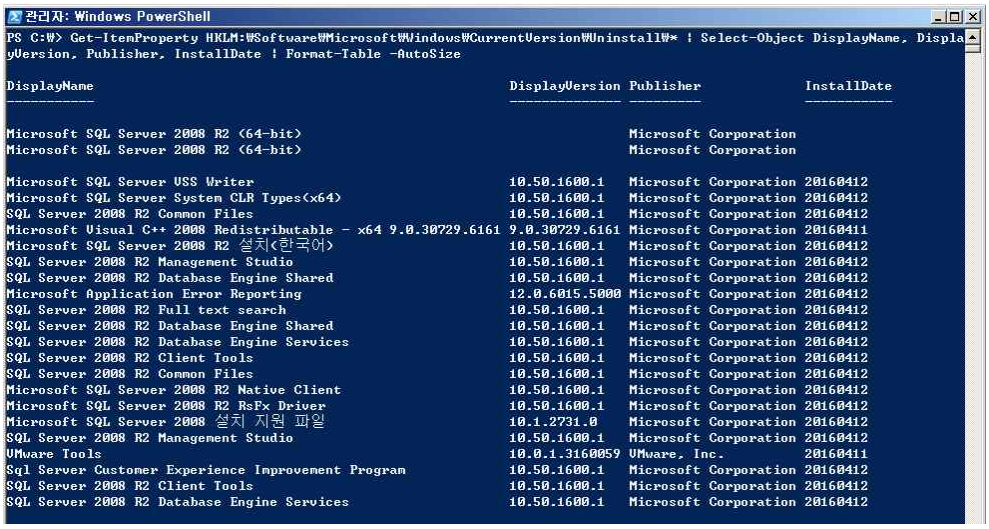
다음은 윈도우 시스템에서 운영체제 설치일시 및 응용프로그램 설치일시를 확인한 예이다.

○ 운영체제 설치일시(윈도우 시스템)

```
C:\Users\Administrator>systeminfo | findstr /I "원래"
원래 설치 날짜:            2016-04-11, 오후 2:55:56
------------------------------------------------------------
------
PS C:\> Get-WmiObject win32_operatingsystem | Select-Object name,cs
name, installdate,lastbootuptime, localdatetime, serialnumber | Format-list

name : Microsoft Windows Server 2008 R2 Datacenter |C:\Windows|\Device\Harddisk0
\Partition1
csname           : WIN-097C9SUDKE5
installdate      : 20160411145556.000000+540
lastbootuptime : 20161012111637.109999+540
localdatetime   : 20161121160828.916000+540
serialnumber     : 00496-001-0001283-84696
```

○ 응용프로그램 설치일시(윈도우 시스템, Registry)



```
관리자: Windows PowerShell
PS C:\> Get-ItemProperty HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\* | Select-Object DisplayName, Displa
yVersion, Publisher, InstallDate | Format-Table -AutoSize

DisplayName                                                    DisplayVersion Publisher            InstallDate
-----------                                                    -------------- ---------            -----------

Microsoft SQL Server 2008 R2 (64-bit)                                         Microsoft Corporation
Microsoft SQL Server 2008 R2 (64-bit)                                         Microsoft Corporation

Microsoft SQL Server VSS Writer                                10.50.1600.1   Microsoft Corporation 20160412
Microsoft SQL Server System CLR Types(x64)                     10.50.1600.1   Microsoft Corporation 20160412
SQL Server 2008 R2 Common Files                                10.50.1600.1   Microsoft Corporation 20160412
Microsoft Visual C++ 2008 Redistributable - x64 9.0.30729.6161 9.0.30729.6161 Microsoft Corporation 20160411
Microsoft SQL Server 2008 R2 설치(한국어)                          10.50.1600.1   Microsoft Corporation 20160412
SQL Server 2008 R2 Management Studio                           10.50.1600.1   Microsoft Corporation 20160412
SQL Server 2008 R2 Database Engine Shared                      10.50.1600.1   Microsoft Corporation 20160412
Microsoft Application Error Reporting                          12.0.6015.5000 Microsoft Corporation 20160412
SQL Server 2008 R2 Full text search                            10.50.1600.1   Microsoft Corporation 20160412
SQL Server 2008 R2 Database Engine Shared                      10.50.1600.1   Microsoft Corporation 20160412
SQL Server 2008 R2 Database Engine Services                    10.50.1600.1   Microsoft Corporation 20160412
SQL Server 2008 R2 Client Tools                                10.50.1600.1   Microsoft Corporation 20160412
SQL Server 2008 R2 Common Files                                10.50.1600.1   Microsoft Corporation 20160412
Microsoft SQL Server 2008 R2 Native Client                     10.50.1600.1   Microsoft Corporation 20160412
Microsoft SQL Server 2008 R2 RsFx Driver                       10.50.1600.1   Microsoft Corporation 20160412
Microsoft SQL Server 2008 설치 지원 파일                           10.1.2731.0    Microsoft Corporation 20160412
SQL Server 2008 R2 Management Studio                           10.50.1600.1   Microsoft Corporation 20160412
VMware Tools                                                   10.0.1.3160059 VMware, Inc.          20160411
Sql Server Customer Experience Improvement Program             10.50.1600.1   Microsoft Corporation 20160412
SQL Server 2008 R2 Client Tools                                10.50.1600.1   Microsoft Corporation 20160412
SQL Server 2008 R2 Database Engine Services                    10.50.1600.1   Microsoft Corporation 20160412
```

나. 고려사항

위와 같은 하드디스크 교체와 관련된 시스템 아티팩츠를 조사하는 과정에서 고려하여야 하는 사항은 다음과 같다.

○ 고스트 등의 백업·복구 솔루션을 사용한 경우에는 별도의 고스트 핑거프린트 등의 조사가 필요하다.

○ 일정 규모 이상의 회사의 경우 이러한 전산작업을 하는 경우 작업계획서를 작성한다. 따라서 하드디스크의 교체 사실을 확인하였다면 그에 대한 작업계획서를 확보하여 한다.

○ 일반적으로 서버를 운영하는 경우 하드디스크는 여러 개를 하나의 RAID 시스템으로 구성하여 사용한다. 이러한 서버를 운용 중에 하드디스크의 장애가 발생하였다면 장애가 발생한 하드디스크만을 교체하게 된다. 이 경우 논리적 볼륨 상에는 변경된 데이터가 없기 때문에 고의적인 RAID로 구성된 전체의 하드디스크 또는 RAID로 구성되지 않은 단일 하드디스크의 교체행위와는 구별하여야 한다.

○ 매번의 현장에서 위에서 제시한 아티팩츠를 전부 조사하고 수집하는 것은 다소 무의미하다. 통상적으로 하드디스크의 포맷과 운영체제의 설치는 동시에 이루어 진다. 둘 중 하나의 아티팩츠만 확인하여도 교체로 의심되는 상황이라고 판단할 수 있다. 따라서 우선순위를 정하여 각 시스템의 특성에 따라 신속하고 효율적으로 하드디스크의 교체 여부를 판단할 수 있는 아티팩츠를 조사한다. 이후 교체로 의심스러운 정황이 확인되면 해당 사실을 뒷받침해 줄 수 있는 다른 아티팩츠를 추가적으로 조사하는 것이 효율적이다.

○ 만약 범죄사실의 발생 시점부터 압수수색 당일의 시점 사이에 이러

한 교체 행위가 일어났거나 압수수색 당일의 시점과 매우 근접하게 하드디스크가 교체되었다면 고의적인 은닉행위라고 판단할 수 있다.

○ 확인할 수 있는 정보가 교체 시간에 대한 정보뿐이지만 서버의 하드 디스크를 교체할 수 있는 사람은 해당 서버의 관리자이기 때문에 별도의 행위자의 특정은 무의미하다.

3. 데이터베이스의 교체

또 다른 은닉행위 유형으로 새로운 데이터베이스를 생성하여 범죄 혐의와 무관한 데이터만을 복원하는 경우가 있다. 데이터베이스를 교체하기 전에 원 데이터베이스에는 기업의 주요 핵심적인 정보가 저장되어 있기 때문에 피 압수자의 입장에서도 중요한 데이터이다. 데이터 은닉을 위해서 단순히 데이터를 삭제하고 새로운 데이터베이스를 생성하여 사용하는 경우는 극히 드물 수밖에 없다. 따라서 데이터 교체 행위와 함께 데이터의 백업도 동반 수행될 수 있다.

데이터베이스의 생성일시, 데이터베이스의 On/Off 일시, 백업 수행 일시 등의 조사를 통하여 데이터베이스의 교체 및 백업 수행 내역을 확인할 수 있다. 이 과정에 조사하여야 할 대상과 고려사항은 다음과 같다.

가. 조사대상

다음 [표 4-3]에서 각 수반행위에 따라 생성되는 아티팩츠를 각 항목에 따라 분류하였다. 각 운영체제와 DBMS에 따라 데이터베이스의 생성일시, 데이터베이스의 On/Off 일시, 백업수행 일시가 확인 가능하다.

번호	절차명	조사대상			
		대분류	중분류	소분류	상세내용
2.1	Database /Table 생성일시	OS	Windows	File system	데이터파일 생성일시
			Linux		데이터파일 생성일시
		DBMS	Oracle	Meta data	데이터파일 생성일시 : v$datafile_header;
			SQL Server	Meta data	▪ mdf, ldf 파일의 생성일자 : sys.master_files, sys.databases ▪ 테이블 생성일자 : sys.objects
			MySQL	Meta data	테이블 생성일자 : INFORMATION_SCHEMA.TABLES
2.3	Backup 수행 이력	OS	Windows	Event Log	응용프로그램 로그 Event ID : 18264(SQLServer 백업) Event ID : 3408(SQLServer 복원)
			linux	Cron Log	Cron 수행내역 : /var/log/cron
				Shell history	/계정홈/.(사용 shell)_history
		DBMS	Oracle	Meta data	RMAN 로그 : V$RMAN_STATUS, V$RMAN_OUTPUT V$rman_backup_job_details
					6초이상 소요되는 작업(백업/복구/통계) : v$session_longops;

<table>
<tr><th rowspan="2">번호</th><th rowspan="2">절차명</th><th colspan="4">조사대상</th></tr>
<tr><th>대분류</th><th>중분류</th><th>소분류</th><th colspan="3">상세내용</th></tr>
<tr><td></td><td></td><td></td><td>SQL Server</td><td>Meta data</td><td colspan="3">백업수행 내역
: msdb.dbo.backupfile
msdb.dbo.backupset
msdb.dbo.backupmediafamily</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Error Log</td><td colspan="3">백업수행 내역</td></tr>
<tr><td rowspan="11">2.1</td><td rowspan="11">Database On/OFF 일시</td><td rowspan="9">OS</td><td rowspan="8">Windows</td><td rowspan="8">Event Log</td><td colspan="3">응용프로그램 로그
: Database On/Off 시간</td></tr>
<tr><td>DBMS</td><td>Event ID</td><td>내용</td></tr>
<tr><td rowspan="3">Oracle</td><td>8</td><td>shutdown</td></tr>
<tr><td>12</td><td>프로세스 종료</td></tr>
<tr><td>5</td><td>프로세스 초기화</td></tr>
<tr><td rowspan="3">SQL Server</td><td>17137</td><td>database 시작</td></tr>
<tr><td>5084</td><td>database 옵션 (off →On) 변경</td></tr>
<tr><td>17147</td><td>SQL Server 종료</td></tr>
<tr><td>MySQL</td><td>100</td><td>DB shutdown / start</td></tr>
<tr><td>Linux</td><td>shell history (일부)</td><td colspan="3">/계정홈/.(사용 shell)_history</td></tr>
<tr><td rowspan="2">DBMS</td><td rowspan="2">Oracle</td><td>Alert Log</td><td colspan="3">Instance startup / shutdown / table space on,off 시간
: Dynamic view(x$dbgalertext)</td></tr>
<tr><td>Data Dictionary</td><td colspan="3">Tablespace Online Time : v$datafile</td></tr>
</table>

번호	절차명	조사대상			
		대분류	중분류	소분류	상세내용
			SQL Server	Error Log	Database On/Off 일시
			MySQL	Error Log	mysqld 데몬 시작, 정지 일시

[표 4-3] 데이터베이스 교체와 관련된 시스템 아티팩츠

현재 운영 중인 DBMS의 메타데이터로부터 데이터베이스의 생성일시를 확인하여 데이터베이스의 교체여부를 판단할 수 있다. 데이터베이스의 생성일시보다 저장된 실제데이터의 작성일시가 이전이라면 데이터베이스의 교체여부를 의심해보아야 한다. 또한 운영체제의 파일시스템에서 해당 데이터가 저장된 데이터베이스 파일의 생성일시를 보충적으로 확인한다.

은닉행위 전 데이터 백업 수행이력에 대해서는 데이터베이스가 생성된 시점과 백업 시점을 비교한다. 데이터베이스 시스템를 종료 및 재시작하거나, 혐의사실과 관련된 데이터베이스만을 오프라인 상태로 변경한 기록은 데이터 백업 사실에 대한 보충적 자료로 사용할 수 있다.

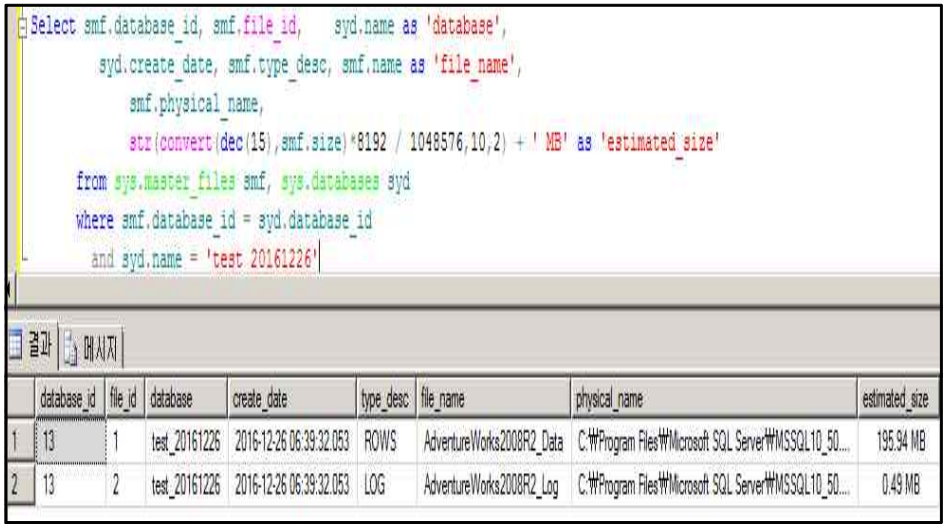
만약 조사 대상 서버가 운영체제로 윈도우 시스템과 DBMS로 SQL-Server를 사용한다면 다음과 같이 시스템 아티팩츠를 확인하여 2016. 12. 26. 06:31:56에 기존 데이터를 백업 후, 2016. 12. 26. 06:34:24에 기존 데이터베이스를 오프라인 한 사실을 확인할 수 있다. 또한 2016. 06:39:33에 새로운 데이터베이스를 생성 후 데이터를 복원한 사실을 확인할 수 있다.

○ 데이터베이스 shutdown / startup / 백업 일시 시간(Error Log)

- 윈도우 이벤트로그 상에도 동일한 내용이 기재 됨.

2016-12-26 06:31:56.01 백업 **Database backed up**. Database: AdventureWorks2008R2
(중략)
2016-12-26 06:34:24.81 spid57 **Setting database option OFFLINE to ON** for database AdventureWorks2008R2.
2016-12-26 06:39:32.97 spid59 Starting up database 'test_20161226'.
2016-12-26 06:39:32.98 spid59 The database 'test_20161226' is marked RESTORING and is in a state that does not allow recovery to be run.
2016-12-26 06:39:33.01 spid59 Starting up database 'test_20161226'.
2016-12-26 06:39:33.18 백업 **Restore is complete on database 'test_20161226'. The database is now available.**
2016-12-26 06:39:33.19 백업 **Database was restored: Database: test_20161226,**
(후략)

○ 데이터 파일의 위치 및 생성시간(SQLServer, Catalog View)

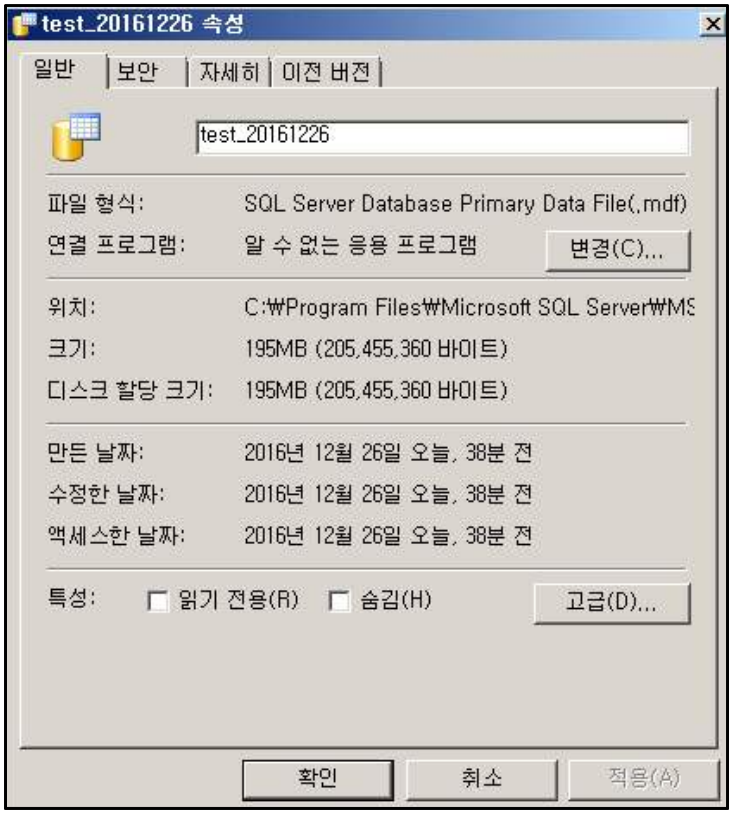


```
Select smf.database_id, smf.file_id,    syd.name as 'database',
       syd.create_date, smf.type_desc, smf.name as 'file_name',
          smf.physical_name,
          str(convert(dec(15),smf.size)*8192 / 1048576,10,2) + ' MB' as 'estimated_size'
    from sys.master_files smf, sys.databases syd
    where smf.database_id = syd.database_id
      and syd.name = 'test_20161226'
```

결과 | 메시지

	database_id	file_id	database	create_date	type_desc	file_name	physical_name	estimated_size
1	13	1	test_20161226	2016-12-26 06:39:32.053	ROWS	AdventureWorks2008R2_Data	C:₩Program Files₩Microsoft SQL Server₩MSSQL10_50....	195.94 MB
2	13	2	test_20161226	2016-12-26 06:39:32.053	LOG	AdventureWorks2008R2_Log	C:₩Program Files₩Microsoft SQL Server₩MSSQL10_50....	0.49 MB

○ 데이터 파일의 위치 및 생성시간(Windows, Filesystem)



나. 고려사항

위와 같은 데이터베이스의 교체와 관련된 시스템 아티팩츠를 조사하는 과정에서 고려하여야 하는 사항은 다음과 같다.

○ 응용프로그램 로그의 이벤트 ID의 경우 중복되거나 고정되지 않을 수 있다.

○ 사실 데이터베이스의 교체 행위를 명확히 판단하는 것은 매우 어렵고 복잡한 일이다. 따라서 새로운 데이터베이스를 생성하였을 경우를 우선적으로 데이터베이스를 교체한 행위로 간주한다. 만약 데이

터베이스를 교체하기 위해서 현재 데이터베이스의 사본을 생성하여 보관하고 원본의 데이터베이스에서 범죄혐의와 관련된 부분을 삭제하였다면 이는 레코드를 삭제한 행위로 보아야 한다. 이 경우 별도 보관 중인 사본을 확보하여 범죄혐의 관련 증거 데이터를 추출한다.

○ 일반적으로 기업은 장애가 발생하였을 경우 그 복구를 위해 데이터베이스의 백업을 주기적으로 수행한다. 위와 같이 확인한 백업이력은 데이터베이스 생성일시와 비교하여 보충적으로 해석하여야 한다.

○ 데이터베이스의 백업은 다른 전산시스템 상의 데이터와 마찬가지로 회사의 정책에 따라 주기적으로 수행하게 된다. 또한 별도의 백업 솔루션을 이용하는 경우가 많다. 이러한 경우 데이터베이스의 생성 시점이 백업솔루션의 설정된 주기 및 회사정책과 일치하는지 여부를 체크하여야 한다.

○ 또한 데이터의 백업 목적이 아니더라도 시스템 패치 작업 등을 위하여 데이터베이스 시스템을 종료하는 경우가 많으므로 해석에 유의하여 한다.

4. 레코드 삭제

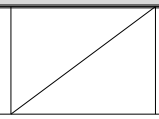
데이터베이스에서 범죄 혐의와 관련된 데이터를 삭제하는 행위는 즉 레코드만을 삭제하는 행위는 상당히 자주 발생할 수 있다. 데이터베이스에서 범죄와 관련된 데이터의 삭제·변경행위를 하고자 하는 사람은 데이터베이스에 접속 후 데이터 변경을 위한 쿼리(DML문)를 통하여 원하는 데이터를 삭제·변경하여야 한다.

이러한 데이터 변경 작업을 수행하였다는 직접적인 정황은 쿼리 로그(일부 시스템에서는 실행계획)와 트랜잭션 로그의 분석을 통해서 확보가 가능하다. 각 시스템의 접속 내역, 트리거 이벤트 등은 이와 같은 사실에 대한 보충적 자료로 사용할 수 있다.

가. 조사대상

다음 [표 4-4]에서 각 수반행위에 따라 생성되는 아티팩츠를 각 항목에 따라 분류하였다. 제시한 운영체제의 시스템로그, DBMS의 메타데이터 및 로그 데이터, 웹 서버의 접속로그 등의 조사를 통해 사용자의 레코드 삭제행위에 대한 식별이 가능하다.

<table>
<tr><th rowspan="2">번호</th><th rowspan="2">절차명</th><th colspan="4">조사대상</th></tr>
<tr><th>대분류</th><th>중분류</th><th>소분류</th><th>상세내용</th></tr>
<tr><td rowspan="8">3.1</td><td rowspan="8">접속내역</td><td rowspan="8">OS</td><td rowspan="8">Windows</td><td rowspan="8">Event Log</td><td colspan="3">보안(Security) 로그, Microsoft-Windows-TerminalServices-LocalSessionManager/Operational 로그</td></tr>
<tr><td rowspan="2">내용</td><td colspan="2">Event ID</td></tr>
<tr><td>evt</td><td>evtx</td></tr>
<tr><td>로그인</td><td>528</td><td>4624</td></tr>
<tr><td>네트워크 로그인</td><td>540</td><td>4624</td></tr>
<tr><td>로그오프</td><td>538</td><td>4634</td></tr>
<tr><td>원격데스크톱 로그온 성공</td><td rowspan="2"></td><td>21</td></tr>
<tr><td>원격데스크톱 로그온 실패</td><td>24</td></tr>
</table>

번호	절차명	조사대상			
		대분류	중분류	소분류	상세내용
					Oracle 접속기록  34
			Linux	System Log	sercure(ssh, telnet 접속로그) : /var/log/secure
					wtmp : 전체 사용자 로그인 내역
					lastlog : 사용자의 마지막 로그인 시간
					pacct : 각 사용자의 명령어 기록
					audit log : 감사로그 (Login Report)
		DBMS	Oracle	Listener Log	접속 기록 : v$diag_alert_ext
			SQL Server	Data Dictionary	현재 세션 연결 기록 : sys.dm_exec_connections sys.dm_exec_sessions
				error Log	성공, 실패 로그인 기록 (default : 실패한 로그인만 기록)
				Audit	감사 정책에 따른 로그인 내역 (default : 사용안함)
			MySQL	General Query Log	접속내역 (접속 ip, 실행 쿼리 기록) : mysql.general_log
3.2	DML문 실행 내역	OS	Linux	shell history	/계정홈/.(사용 shell)_history
		DBMS	Oracle	Redo Log	데이터 변경 내역 조회 : v$logmnr_contents
				Trigger	트리거 리스트 추출 : SYS.ALL_OBJECTS SYS.ALL_TRIGGERS

번호	절차명	조사대상			
		대분류	중분류	소분류	상세내용
			SQL Server	Data Dictionary	실행 계획 캐쉬(Plan cache) : sys.dm_exec_query_stats, sys.dm_exec_sql_text
				Redo Log	데이터 변경 내역 조회 : fn_dblog()
				Trigger	트리거 리스트 추출 : sys.triggers
			MySQL	General Log	쿼리 실행 내역(default : 사용안함) : mysql.general_log
				Redo Log (binary log)	데이터 변경 내역 조회 : /var/lib/mysql/mysql-bin.######
				Trigger	트리거 리스트 추출 : information_schema.TRIGGERS
		Web Server	Apache	Access Log	/var/log/httpd/
			IIS		%SystemDrive%\inetpub\logs\LogFiles\W3SVC#[사이트 ID]\

[표 4-4] 레코드 삭제와 관련된 시스템 아티 팩츠

운영체제의 시스템 로그를 통해서는 운영체제의 터미널에 접속한 내역, 로그인 일시, 명령어를 수행한 내역 등의 정보를 얻을 수 있다.

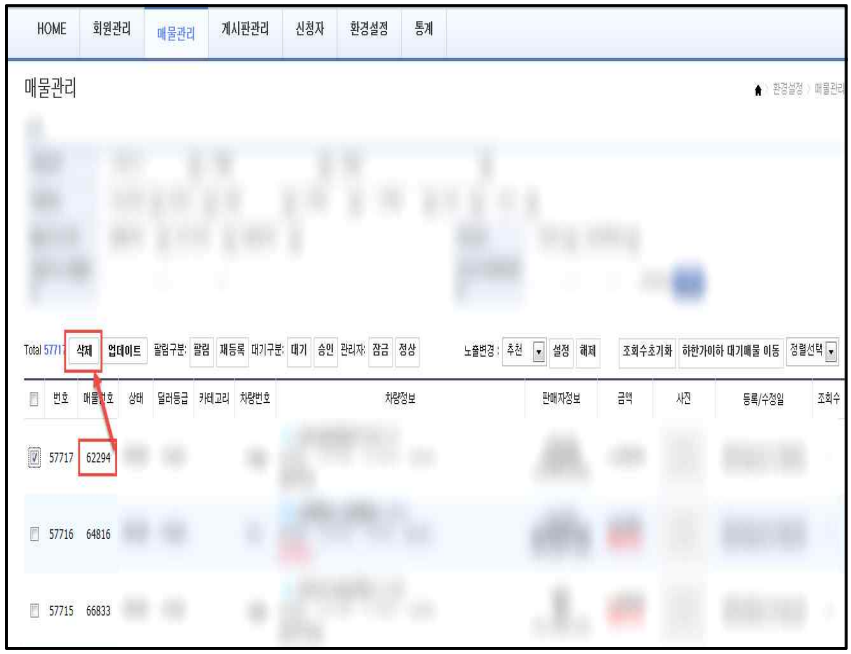
각 데이터베이스 관리시스템(DBMS)의 메타데이터와 트랜잭션 로그, 트리거 이벤트에서는 레코드 삭제를 위한 데이터 변경 명령(DML)문의 실행에 관한 정보, 데이터베이스에 접속한 내역 등을 확인할 수 있다.

웹 서버의 접속로그의 경우 클라이언트(사용자 PC의 웹브라우저)에서

요청한 데이터와 이를 서버에서 제공한 정보와 그에 대한 상태 메시지만을 저장한다. 로그에 기록된 요청한 데이터는 해당 웹 사이트 소스의 이름과 전달된 변수의 정보만을 저장하므로 레코드 삭제 명령을 수행하는 웹 소스와 연관하여 분석하여야 한다.

만약 조사대상 시스템이 웹서버로 Apache, DBMS로 MySQL을 사용한다면 다음과 같이 시스템 아티팩츠를 조사하여 레코드의 삭제 행위를 식별할 수 있다. 사용자에게 제공되는 관리자 ID를 통하여 매물번호 62294에 해당하는 데이터를 삭제하였다면, 2016. 11. 27. 00:02:25에 웹 서버의 접속로그에서 데이터베이스에서 해당 데이터를 삭제하는 명령을 수행하는 웹 소스의 요청한 사실과 2016. 11. 27. 26. MySQL의 쿼리 로그에 기록된 내용을 통해서 ID : jjm223을 통해 접속 후 delete 명령을 수행하였음을 확인할 수 있다. 이와 동시에 MySQL의 binary로그에도 delete 명령을 수행한 흔적을 확인할 수 있다.

(1) 사용자 UI환경에서 매물번호 62294번의 데이터를 삭제



(2) 웹 서버 접속로그 확인

tail -f /var/log/httpd/access_log	
로그 항목	설명
192.168.171.1	클라이언트 IP주소
-	
-	
[27/Nov/2016:00:02:25 +0900]	요청한 처리를 마친 시간
"GET /Mgn/car/regist_ok.html?mode=multi_delete&duid=62294;	클라이언트가 요청한 첫 라인
200	상태코드
181	
"http://192.168.171.129/list.php"	참조 요청 헤더
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)"	클라이언트 브라우저 식별정보

(3) 요청한 소스페이지 확인

[list.php]

```
function sel_delete()
{
    var que = chk_check();
    (생략)
    processIframe.location.href = "regist_ok.html?mode=multi_delete
&duid="+que;
}
```

[regist.html]

```
$mode = $_POST['mode'];
if(!$mode) $mode = $_GET['mode'];

$duid = $_POST['duid'];
if(!$duid) $mode = $_GET['duid'];

(생략)

if($_GET[mode] == "multi_delete")
{
    $darr = explode(';' , $duid);
    (생략)
    for ($i = 0; $i < sizeof($darr); $i++)

    $delQuery = "DELETE FROM in_sales WHERE m_no='{$no}' AND
m_id='".$cm_id."'";

    mysql_query($delQuery);

    $delQuery = "DELETE FROM in_salesAddInfo WHERE i_no='{$no}' AND
i_mId='".$cm_id."'";

    mysql_query($delQuery);

    $delInterQuery = "DELETE FROM interestCar WHERE m_no='
{$no}'";

    mysql_query($delInterQuery);

    (생략)
    exit;
}
```

(4) MySQL의 쿼리 로그 확인

```
SQL> select * from mysql.general_log
        where argument like '%delete%'
```

event_time *	user_host *	thread_id *	server_id *	command_type *	argument *
2016-11-26 오후 11:59:04	root[root] @ [192.168.171.1]	13	1	Query	select * from mysql.general_log where argument like '%delete%'
2016-11-27 오전 12:02:46	jjm223[jjm223] @ localhost []	94	1	Query	DELETE FROM in_sales WHERE m_no='62294'
2016-11-27 오전 12:02:46	jjm223[jjm223] @ localhost []	94	1	Query	DELETE FROM in_salesAddInfo WHERE i_no='62294' AND i_mId='car1015'
2016-11-27 오전 12:02:46	jjm223[jjm223] @ localhost []	94	1	Query	DELETE FROM interestCar WHERE m_no='62294'
2016-11-27 오전 12:02:46	jjm223[jjm223] @ localhost []	94	1	Query	insert into traceWork (tw_mId, tw_ip, tw_seq, tw_mode, tw_action, tw_referer,

(5) MySQL의 binary log 확인

```
# mysqlbinlog /var/lib/mysql/mysql-bin.000008
(중략)
#161127 0:02:46 server id 1 end_log_pos 6175 Query thread_id=94 exec_time=0 error_code=0
SET TIMESTAMP=1480172566/*!*/;
DELETE FROM in_sales WHERE m_no='62294'
/*!*/;
# at 6175
#161127 0:02:46 server id 1 end_log_pos 6306 Query thread_id=94 exec_time=0 error_code=0
SET TIMESTAMP=1480172566/*!*/;
DELETE FROM in_salesAddInfo WHERE i_no='62294' AND i_mId='car1015'
/*!*/;
# at 6306
#161127 0:02:46 server id 1 end_log_pos 6413 Query thread_id=94 exec_time=0 error_code=0
SET TIMESTAMP=1480172566/*!*/;
DELETE FROM interestCar WHERE m_no='62294'
/*!*/;
# at 6413
(중략)
```

이 외 기타 시스템에서 접속기록 및 삭제 명령을 수행한 확인한 예시를 다음과 같다.

○ 원격 데스크탑(RDP) 접속기록(윈도우시스템, 이벤트로그)

```
PS C:\> Get-WinEvent Microsoft-Windows-TerminalServices-LocalSessionManager/Operational | Where-Object {$_.ID -eq "21"} | Format-List

TimeCreated  : 2016-11-14 오후 3:19:50
ProviderName : Microsoft-Windows-TerminalServices-LocalSessionManager
Id           : 21
Message      : 원격 데스크톱 서비스: 세션 로그온 성공:
               사용자: WIN-097C9SUDKE5\jjm223
               세션 ID: 2
               원본 네트워크 주소: 192.168.171.1
```

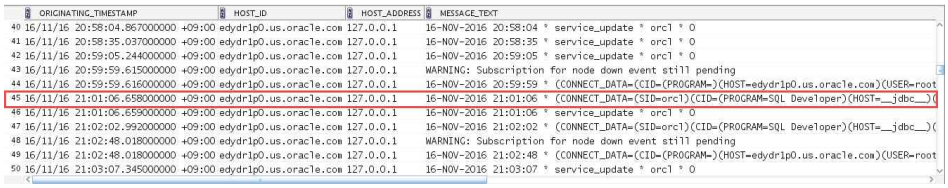
○ 오라클 접속 기록(윈도우시스템, 이벤트로그)

○ 운영체제 터미널 접속 기록(리눅스시스템, secure log)

[root@localhost etc]# cat /var/log/secure
(중략)
Oct 21 17:39:32 localhost sshd[92321]: Accepted password for root from 192.168.171.1 port 57510 ssh2
Oct 21 17:39:32 localhost sshd[92321]: pam_unix(sshd:session): session opened for user root by (uid=0)
Oct 21 17:39:50 localhost sshd[92321]: pam_unix(sshd:session): session closed for user root
(후략)

○ 데이터베이스 접속 기록(오라클, Alert Log)

```
SQL>
select ORIGINATING_TIMESTAMP,HOST_ID,HOST_ADDRESS,MESSAGE_TEXT,FILENAME
from V$DIAG_ALERT_EXT
where ORIGINATING_TIMESTAMP
      between TO_TIMESTAMP('2016-11-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss')
      and TO_TIMESTAMP('2016-11-30 23:59:59', 'yyyy-mm-dd hh24:mi:ss')
and TRIM(COMPONENT_ID)='tnslsnr';
```



	ORIGINATING_TIMESTAMP	HOST_ID	HOST_ADDRESS	MESSAGE_TEXT
40	16/11/16 20:58:04.867000000 +09:00	edydr1p0.us.oracle.com	127.0.0.1	16-NOV-2016 20:58:04 * service_update * orcl * 0
41	16/11/16 20:58:35.037000000 +09:00	edydr1p0.us.oracle.com	127.0.0.1	16-NOV-2016 20:58:35 * service_update * orcl * 0
42	16/11/16 20:59:05.244000000 +09:00	edydr1p0.us.oracle.com	127.0.0.1	16-NOV-2016 20:59:05 * service_update * orcl * 0
43	16/11/16 20:59:59.615000000 +09:00	edydr1p0.us.oracle.com	127.0.0.1	WARNING: Subscription for node down event still pending
44	16/11/16 20:59:59.616000000 +09:00	edydr1p0.us.oracle.com	127.0.0.1	16-NOV-2016 20:59:59 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=edydr1p0.us.oracle.com)(USER=root
45	16/11/16 21:01:06.658000000 +09:00	edydr1p0.us.oracle.com	127.0.0.1	16-NOV-2016 21:01:06 * (CONNECT_DATA=(SID=orcl)(CID=(PROGRAM=SQL Developer)(HOST=__jdbc__)(
46	16/11/16 21:01:06.659000000 +09:00	edydr1p0.us.oracle.com	127.0.0.1	16-NOV-2016 21:01:06 * service_update * orcl * 0
47	16/11/16 21:02:02.992000000 +09:00	edydr1p0.us.oracle.com	127.0.0.1	16-NOV-2016 21:02:02 * (CONNECT_DATA=(SID=orcl)(CID=(PROGRAM=SQL Developer)(HOST=__jdbc__)(
48	16/11/16 21:02:48.018000000 +09:00	edydr1p0.us.oracle.com	127.0.0.1	WARNING: Subscription for node down event still pending
49	16/11/16 21:02:48.018000000 +09:00	edydr1p0.us.oracle.com	127.0.0.1	16-NOV-2016 21:02:48 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=edydr1p0.us.oracle.com)(USER=root
50	16/11/16 21:03:07.345000000 +09:00	edydr1p0.us.oracle.com	127.0.0.1	16-NOV-2016 21:03:07 * service_update * orcl * 0

▶ **16-NOV-2016 20:48:10 * (CONNECT_DATA=(SID=orcl)(CID=(PROGRAM=SQL Developer)(HOST=__jdbc__)(USER=oracle))) *(ADDRESS=(PROTOCOL=tcp)(HOST=127.0.0.1)(PORT=9252)) * establish * orcl * 12505**

○ 삭제 명령어 수행 내역(SQL-Server, Plan Cache)

```
SQL>
select sqlText.text,
        queryStats.last_execution_time,
        queryStats.creation_time,
        queryStats.execution_count,
        queryStats.total_elapsed_time
from sys.dm_exec_query_stats as queryStats
cross apply sys.dm_exec_sql_text(queryStats.plan_handle) as sqlText
where sqlText.text like '%delete%'
order by queryStats.last_execution_time desc
```

text	last_execution_time *	creation_time *	execution_count *	total_elapsed_time *
select sqlText.text, queryStats.last_execution_time, queryStats.cr...	2016-11-24 오후 3:35:48	2016-11-24 오후 3:35:48	1	12338
select a.session_id, a.connect_time, a.last_read, a.last_write, b.[host_name...	2016-11-24 오후 3:17:29	2016-11-24 오후 3:17:29	1	85323
select a.session_id, a.connect_time, a.last_read, a.last_write, b.[host_name...	2016-11-24 오후 3:17:29	2016-11-24 오후 3:17:29	1	503
(@1 varchar(8000))DELETE [test20161117].[dbo].[test] WHERE [a1]=@1	2016-11-23 오전 2:21:31	2016-11-23 오전 2:21:31	1	331

▶ (@1 varchar(8000))DELETE [test20161117].[dbo].[test] WHERE [a1]=@1

○ 트랜잭션 로그 기록(Oracle, redo log)

Oracle Logminer 조사 예시

```
1. Redo logs 상태/파일 확인
   select * from v$logfile;
   select * from v$log;

2. utl_file_dir 위치 확인
   show parameter utl;
3. utl_file_dir 위치에 Dictionayr file 생성
   exec dbms_logmnr_d.build(dictionary_filename=>'dict2.dat', dictionary_loca
   tion=>'/app/oracle/logminer');
```

4. 현재 사용 중인 Redo log 파일 추가

```
exec dbms_logmnr.add_logfile('/u01/app/oracle/oradata/orcl/redo02.log',1);
```

5. 추가된 파일 확인

```
select * from v$logmnr_logs;
```

6. timestamp 형식 변경

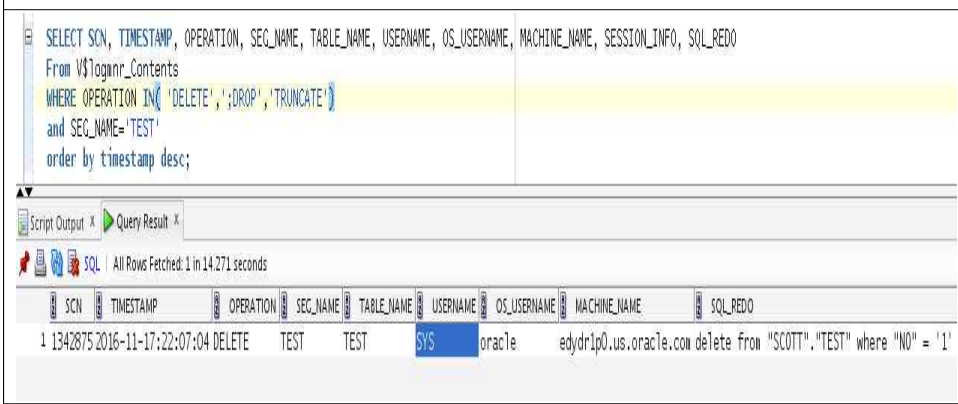
```
alter session set nls_date_format='YYYY-MM-DD:HH24:MI:SS';
```

7. dict2.dat에 대한 logminer 시작

```
exec dbms_logmnr.start_logmnr(dictfilename=>'/app/oracle/logminer/dict2.dat');
```

8. v$logmnr_contents에서 내용 조회

```
SELECT SCN, TIMESTAMP, OPERATION, SEG_NAME,
       TABLE_NAME, USERNAME, OS_USERNAME,
       MACHINE_NAME, SESSION_INFO, SQL_REDO
From V$logmnr_Contents
WHERE OPERATION IN( 'DELETE','DROP','TRUNCATE')
  AND SEG_NAME='TEST';
```



○ 저장 트리거 내역 조회(SQLServer, 시스템테이블)

```
SELECT obj.object_id as [ObjectId],
        object_name( trg.parent_id ) as ParentName,
        obj.name as [Name],
        trg.create_date as [CreateDate],
        ppt.value as [Comments],
        trg.is_disabled,
        com.text
FROM sys.objects obj
INNER JOIN sys.triggers AS trg
    ON trg.object_id = obj.object_id
INNER JOIN sys.syscomments com
    ON com.id = obj.object_id
LEFT OUTER JOIN sys.extended_properties AS ppt
  ON obj.object_id = ppt.major_id
  AND ppt.minor_id = 0
  AND ppt.name = 'MS_Description'
WHERE obj.type = N'TR'
```

	ObjectId	ParentName	Name	CreateDate	Comments	is_disabled	text
1	1483152329	Employee	dEmployee	2012-03-28 13:52:34.453	INSTEAD OF DELETE trigger which keeps ...	0	CREATE TRIGGER [HumanResources].[dEmployee] O...
2	1499152386	Person	iuPerson	2012-03-28 13:52:34.477	AFTER INSERT, UPDATE trigger inserting i...	0	CREATE TRIGGER [Person].[iuPerson] ON [Person].[Pe...
3	1515152443	PurchaseOrderDetail	iPurchaseOrderDetail	2012-03-28 13:52:34.483	AFTER INSERT trigger that inserts a row in t...	0	CREATE TRIGGER [Purchasing].[iPurchaseOrderDetail]...
4	1531152500	PurchaseOrderDetail	uPurchaseOrderDetail	2012-03-28 13:52:34.487	AFTER UPDATE trigger that inserts a row in ...	0	CREATE TRIGGER [Purchasing].[uPurchaseOrderDetail...
5	1547152557	PurchaseOrderHeader	uPurchaseOrderHeader	2012-03-28 13:52:34.490	AFTER UPDATE trigger that updates the Re...	0	CREATE TRIGGER [Purchasing].[uPurchaseOrderHead...
6	1563152614	SalesOrderDetail	iduSalesOrderDetail	2012-03-28 13:52:34.497	AFTER INSERT, DELETE, UPDATE trigger ...	0	CREATE TRIGGER [Sales].[iduSalesOrderDetail] ON [S...
7	1579152671	SalesOrderHeader	uSalesOrderHeader	2012-03-28 13:52:34.517	AFTER UPDATE trigger that updates the Re...	0	CREATE TRIGGER [Sales].[uSalesOrderHeader] ON [S...
8	1595152728	Vendor	dVendor	2012-03-28 13:52:34.520	INSTEAD OF DELETE trigger which keeps ...	0	CREATE TRIGGER [Purchasing].[dVendor] ON [Purcha...
9	1611152785	WorkOrder	iWorkOrder	2012-03-28 13:52:34.523	AFTER INSERT trigger that inserts a row in t...	0	CREATE TRIGGER [Production].[iWorkOrder] ON [Prod...
10	1627152842	WorkOrder	uWorkOrder	2012-03-28 13:52:34.527	AFTER UPDATE trigger that inserts a row in ...	0	CREATE TRIGGER [Production].[uWorkOrder] ON [Pro...

나. 고려사항

위와 같은 레코드 삭제와 관련된 시스템 아티팩츠를 조사하는 과정에서 고려하여야 하는 사항은 다음과 같다.

○ 일반적으로 데이터베이스에서 혐의 사실을 은닉하기 위해서 레코드를 수정(update)하는 경우는 드물고 이러한 경우 발견이 어렵다. 통상적인 시스템에서는 권한이 있는 사용자는 수시로 데이터의 변경을 시도한다. 또한 범죄와 관련 된 몇 개의 레코드만을 변경하는 것이 아니라면 관련된 다른 레코드 또는 항목 값들이 이러한 레코드를 참조하기 때문에 해당 데이터만을 변경할 수는 없다.

○ 트랜잭션 로그를 압수수색 현장에서 조사는 경우 그 용량 및 서버 자원의 사용량 등을 체크하여 수행하여야 한다. 운영 중인 시스템상에서 상당히 큰 용량의 트랜잭션 로그의 분석을 시도 할 경우 시스템 과부하가 발생하여 서버 운영의 지장을 초래할 수 있다.

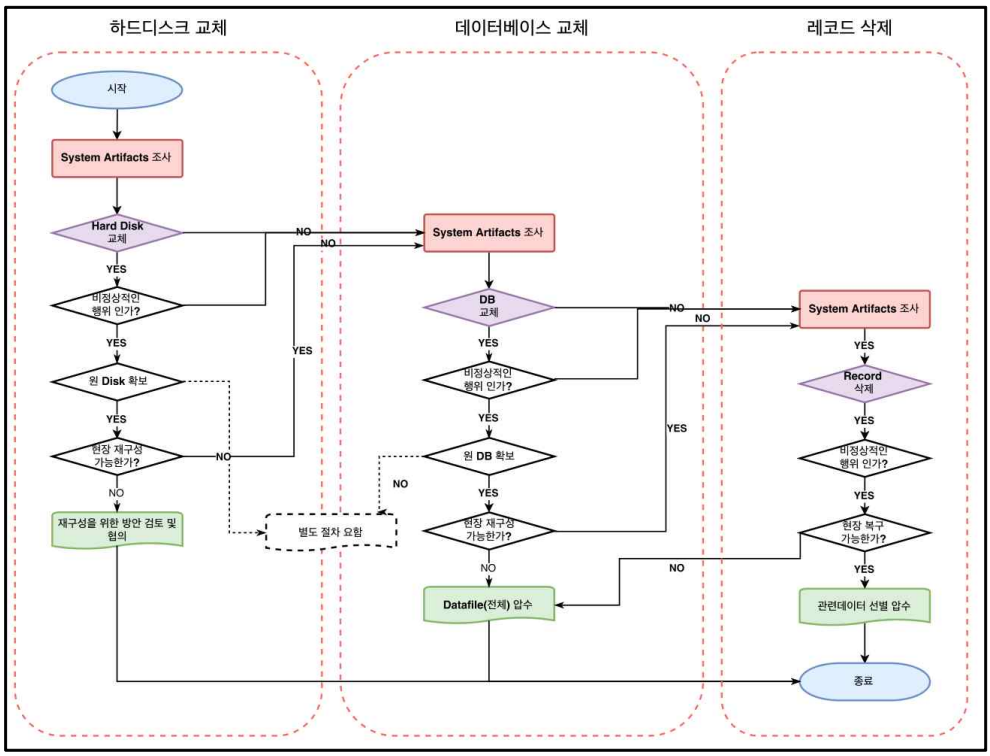
○ 데이터베이스는 성능을 목적으로 미리 수행하여할 작업을 옵티마이저에 의해 실행계획을 작성한 후 실행한다. 이렇게 세운 실행계획을 캐쉬로 저장하고 있다가 새로운 작업이 할당되면 이러한 캐쉬와 비교하여 동일한 경우 기존에 작성하였던 실행계획을 이용한다. 이러한 실행계획 상에 레코드 삭제를 위한 명령어가 저장되어 있을 수 있다. 하지만 이러한 실행계획은 그 보관 주기가 짧고 공통적인 요소로 자동 분류되어 저장되므로 특정테이블의 특정레코드를 삭제를 위한 명령어를 발견하기 어려운 경우가 있다.

○ 각 DBMS의 설치 시 디폴트로 쿼리에 대한 감사를 설정하지 않는다. 압수수색 현장에서 이러한 감사 설정을 확인하여 실행한 쿼리에 대한 정보가 별도의 감사 기록에 기록되는지 확인하여야 한다.

Ⅴ. 조사 기법의 활용

1. 단계별 압수수색 절차모델

4장에서 데이터의 물리적인 저장구조로 인하여, 은닉행위 유형에 따라 압수수색 절차를 구분하여 단계적으로 진행 할 수 있다고 설명하였다. 다음 [그림 5-1]과 같이 데이터 은닉행위 발생 시, 압수수색 진행 과정을 행위 유형에 따라 각 단계로 구분하여 진행할 수 있다. 각 단계 별 해당 아티팩츠를 조사 후 은닉행위의 발생 유무를 확인하여 다음 단계로 진행하게 된다.



[그림 5-1] 은닉행위에 따른 압수수색 절차 진행 모델

우선 하드디스크의 교체 여부를 판단하고 교체 유무에 따라 조사 절차를 종료할 것인지 결정하다. 만약 하드디스크가 교체되었다면 교체 전의 하드디스크의 확보를 위한 시도를 하여야 한다. 하드디스크가 확보되었을 경우 그것을 현장에서 재구성하여 데이터를 추출할 것인지에 대한 결정한다.

하드디스크가 교체되지 않았다면 데이터베이스의 교체 여부를 판단하게 된다. 데이터베이스의 교체 유무에 따라 조사 절차를 종료할 것인지 결정한다. 데이터베이스가 교체되었을 경우 교체 전의 데이터베이스를 확보하기 위한 시도를 하여야 한다. 교체 전의 데이터베이스가 확보되었다면 현장에서 시스템을 재구성하여 데이터를 추출할 것인지 결정한다.

하드디스크와 데이터베이스가 모두 정상이라면 레코드의 삭제 행위가 있었는지를 조사한다. 만약 레코드가 삭제된 정황이 발견되었다면, 현장에서 삭제된 레코드의 복구 작업이 가능한지 판단한다.

위와 같은 압수수색 절차를 진행을 통하여, 디지털포렌식 수사관은 상황에 대한 신속하고 명확한 판단을 할 수 있으며 압수수색 전 과정을 보다 효율적으로 진행할 수 있을 것이다.

2. 쉘스크립트를 이용한 조사 방법

효율적인 시스템아티팩츠의 수집을 위하여 다음과 같이 윈도우의 파워쉘을 활용한 일괄 추출 스크립트를 제시한다. 앞서 제시한 수많은 아티팩츠를 각 항목 별로 조회하는 것은 비효율적이다. 운영체제에 따라 리눅스 시스템의 경우 쉘스크립트를, 윈도우 시스템의 경우 파워셀 스크립트을 이용하여 관련 항목의 데이터를 일괄 추출할 수 있다. 다음은 윈도우시스템의 레지스트리와 이벤트로그, SQLServer의 메타데이터와 로그

에서 수반 행위에 대한 시스템 아티팩츠를 일괄 추출하는 파워셸 스크립트의 예이다.

○ 사양 : Windows server 2008, SQLServer 2008

```
--------------------------------------------------------------------------------

start-transcript work_log.txt

echo "----- 1.2 Os installed date ----"
echo ""
echo "-----        Regsitry            ----"
echo ""
Get-ItemProperty        Registry::"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion" | Select-Object productname, productid, installdate | Format-List
echo "-----             WMI              -----"
echo ""
Get-WmiObject win32_operatingsystem |
Select-Object     name,csname,installdate,lastbootuptime,     localdatetime,     serialnumber     |
Format-list

echo "-- 1.3 Application Install date -"
echo ""
echo "--             Regsitry           ------"
echo ""
Get-ItemProperty
HKLM:\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\*                  |
Select-Object DisplayName, Publisher, InstallDate | Format-Table -AutoSize
Get-ItemProperty        HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\*       |
Select-Object DisplayName, Publisher, InstallDate | Format-Table -AutoSize
echo "--             Event Log          ------"
echo ""
echo "-----        event log config    ----"
Get-WinEvent     -listlog     system,security,application     |     format-list     -property
logname,filesize,lslogfull,oldestrecordnumber,recordcount,isenabled,MaximumSizeInBytes,LogFile
Path,LogMode
Get-EventLog Application | Where-Object {$_.EventID -eq "1033"} | Format-List >
App_InstallDate.txt
echo "----created App_IntallDate.txt----"
echo ""
echo "----     2.1 DB on/off date     ----"
echo ""
```

```
echo "-----          event log          ----"
echo ""
Get-EventLog Application | Where-Object {$_.EventID -eq "17137" -or $_.EventID -eq "5084" -or $_.EventID -eq "17147"} | Format-List * > DB_OnOffDate.txt
Get-EventLog Application | Where-Object {$_.EventID -eq "17137" -or $_.EventID -eq "5084" -or $_.EventID -eq "17147"} | Format-table -autosize -property index,eventid, timegenerated, Machinename,message  > DB_OnOffdateList.txt
echo "----created DB_OnOffDate.txt  -----"
echo ""
echo "----created DB_OnOffdateList.txt----"
echo ""

echo "----     2.2 datfile createdtime  ---"
echo ""
echo "-----            file system          ---"
echo ""
Get-ChildItem 'C:\Program Files\Microsoft SQL Server' -recurse -include *.mdf,*.ldf | Format-list -property fullName,CreationTime,LastWriteTime,LastAccessTime > datafile_mactime.txt
echo "-------created datafile_mactime.txt--"
echo ""
echo "-----   SQLServer Catalog view   ----"
echo ""
Invoke-Sqlcmd -Query "select name,crdate,filename from sys.sysdatabases" | Format-List > SQLSever_mdf_list.txt
echo "----    created SQLServer_mdf_list.txt-"
echo ""

echo "----    2.3 backup excuted date      ---"
echo ""
Invoke-Sqlcmd -Query "
select a.file_size,b.name, a.logical_name, a.physical_name,b.database_creation_date, b.backup_start_date, b.backup_finish_date, b.database_name, b.type, c.physical_device_name
from msdb.dbo.backupfile a
join msdb.dbo.backupset b
on a.backup_set_id = b.backup_set_id
join msdb.dbo.backupmediafamily c
on a.backup_set_id = c.media_set_id" |export-csv Backup_excuted_list.csv
echo "-----  creatd Backup_excuted_list.csv -"
echo ""
Get-EventLog Application | Where-Object {$_.EventID -eq "18264" -or $_.EventID -eq "3408"} | Format-List * > Backup_excuted_list.txt
echo "-----  creatd Backup_excuted_list(EventLog).txt  ---"
```

```
echo ""

echo "-----          3.1 Access History         ---"
echo ""
echo "-----       OS Login/off(eventLog)      ---"
echo ""
Get-EventLog Security | Where-Object {$_.EventID -eq "4624" -or $_.EventID -eq "4634"}
| Format-List > login_off.txt
echo "-----         creatd Login_off.txt       ---"
echo ""
echo "-----                RDP Login                ----"
echo ""
Get-WinEvent     Microsoft-Windows-TerminalServices-LocalSessionManager/Operational     |
Where-Object {$_.ID -eq "21"} | Format-List > RDP_loginoff_list.txt
echo "-----         RDP_loginoff_list.txt      -----"
echo ""
echo "-----         Current session Login      -----"
echo ""
Invoke-Sqlcmd -Query "
select a.session_id, a.connect_time,
        a.last_read,a.last_write,
        b.[host_name], b.[program_name],
        b.login_name, b.nt_domain,
        b.nt_user_name, a.net_transport,
        a.auth_scheme, a.protocol_type,
        a.client_net_address,
        a.client_tcp_port,
        a.local_net_address, a.local_tcp_port,a.endpoint_id, b.[status], b.row_count
from sys.dm_exec_connections a
join sys.dm_exec_sessions b
on a.session_id = b.session_id" > Current_session_Login.txt
echo "-----    created Current_session_Login.txt   ---"
echo ""

echo "-----  DML DDL statement excuted list  ----"
echo ""
Invoke-Sqlcmd -Query "
use test20161117;
SELECT
[Current LSN],
[Transaction ID],
 [Operation],
 [Transaction Name],
```

```
  [CONTEXT],
  [AllocUnitName],
  [Page ID],
  [Slot ID],
  [Begin Time],
  [End Time],
  [Number of Locks],
  [Lock Information]
FROM sys.fn_dblog(NULL,NULL)
WHERE [Transaction ID] in (
            SELECT
             [Transaction ID]
            FROM sys.fn_dblog(NULL,NULL)
                        WHERE Operation IN
                           ('LOP_MODIFY_ROW',
                                 'LOP_DELETE_ROWS')          )"          |          export-csv
DML_excuted_list_ldf.csv
echo "-----  created DML_excuted_list_ldf.csv  -"

echo "-----              trigger list                -"
Invoke-Sqlcmd -Query "
use AdventureWorks2008R2;
SELECT obj.object_id as [ObjectId],
         object_name( trg.parent_id ) as ParentName,
         obj.name as [Name],
         trg.create_date as [CreateDate],
         ppt.value as [Comments],
         trg.is_disabled,
         com.text
   FROM sys.objects obj
    INNER JOIN sys.triggers AS trg
         ON trg.object_id = obj.object_id
    INNER JOIN sys.syscomments com
         ON com.id = obj.object_id
    LEFT OUTER JOIN sys.extended_properties AS ppt
         ON obj.object_id = ppt.major_id
         AND ppt.minor_id = 0
         AND ppt.name = 'MS_Description'
  WHERE obj.type = N'TR'" > trigger_list.txt
echo "-----created trigger_list.txt-----"

stop-transcript
```

위 파워셀 스크립트를 이용하여 다음과 같이 윈도우 설치일시, 응용프로그램 설치일시, 데이터베이스 온/오프 일시, 데이터파일 생성시간, 백업 수행 이력, 운영체제 로그인/로그오프 일시, 원격접속 로그인 기록, 현재 세션의 연결 정보, 트랜잭션 로그의 DML 문 실행 내역, 저장된 트리거 이벤트 내역을 일괄하여 추출가능하다.

이름	수정한 날짜	유형	크기
a	2016-11-28 오전 ...	PS1 파일	8KB
App_InstallDate	2016-11-28 오전 ...	텍스트 문서	107KB
Backup_excuted_list	2016-11-28 오전 ...	CSV 파일	2KB
Backup_excuted_list	2016-11-28 오전 ...	텍스트 문서	12KB
Current_session_Login	2016-11-28 오전 ...	텍스트 문서	13KB
datafile_mactime	2016-11-28 오전 ...	텍스트 문서	13KB
DB_OnOffDate	2016-11-28 오전 ...	텍스트 문서	108KB
DB_OnOffdateList	2016-11-28 오전 ...	텍스트 문서	25KB
DML_excuted_list_ldf	2016-11-28 오전 ...	CSV 파일	1KB
login_off	2016-11-28 오전 ...	텍스트 문서	1,220KB
RDP_loginoff_list	2016-11-28 오전 ...	텍스트 문서	6KB
SQLSever_mdf_list	2016-11-28 오전 ...	텍스트 문서	4KB
trigger_list	2016-11-28 오전 ...	텍스트 문서	55KB
work_log	2016-11-28 오전 ...	텍스트 문서	26KB

Ⅵ. 결론

지금까지 데이터베이스와 관련된 디지털 포렌식 연구 분야에서는 삭제된 레코드의 복구, 데이터베이스의 구조 분석을 통한 일반화된 분석 프로세스, 각 제조사 별 특성에 따른 세부적인 분석 방법 등의 많은 연구가 진행되어 왔다. 그러나 광범위한 시스템이 존재하는 압수수색 현장에서 발생하는 다양한 상황에 이러한 연구 결과들을 적용하기에는 어려움이 있다. 따라서 사건 현장에서 발생하는 상황에 따른 구체적인 대응 방법이 마련되어야 한다. 사전에 수많은 장비와 프로그램 중에서 범죄사실과 관련된 증거를 저장하거나 사용자의 행위를 판단할 수 있는 항목들에 대한 정의 및 선정을 통해 이에 대한 대응 방법에 따라 압수수색을 진행하는 것이 효율적이다.

데이터베이스의 압수 시, 사용자의 행위 분석이 요구되는 주요한 부분은 데이터베이스의 은닉행위 수행여부이다. 데이터베이스 은닉행위는 여러 관련 시스템 아티팩츠의 조사를 통하여 판단할 수 있다. 압수수색 현장에서 은닉행위가 발생하였을 경우 시스템 아티팩츠의 조사를 통하여 범죄 혐의사실이외의 데이터가 포함된 전체 데이터베이스 파일의 판단 근거 및 추가적인 증거인멸 혐의에 대한 증거의 수집이 가능하다. 이러한 사전에 은닉행위와 관련된 시스템 아티팩츠를 선정하여 압수수색 전 과정에 대한 구체적인 절차를 정의함으로써 효율적인 압수수색을 진행할 수 있다.

본 논문에서는 압수수색 현장에서 범죄 혐의와 관련된 데이터베이스에 저장된 데이터를 피 압수자가 은닉하는 상황이 발생하였을 경우, 시스템 아티팩츠의 조사를 통해서 그러한 은닉행위를 식별하는 방법에 대해 고찰하였다. 이 과정에서 네트워크로 연결된 다양한 장비들 중에서 은닉행

위와 관련된 DBMS, 웹 서버와 여기에 기반이 되는 운영체제 세 가지로 선별하였다. 다음으로 이러한 시스템에 설치되어 운영될 수 있는 수많은 프로그램 중에서 시장 점유율를 기초로 현장에서 접할 확률이 높은 프로그램을 선정하여 본 논문의 연구대상을 삼았다. 그리고 피 압수자의 은닉행위를 하드디스크 교체, 데이터베이스 교체, 레코드 삭제의 세 가지 유형으로 구분하여 정의하였다. 그리고 각 유형 별로 은닉행위자가 수행하게 되는 수반 행위 따라 생성되는 시스템 아티팩츠를 선정하고 이에 대한 조사 방법에 대하여 고찰하였다. 이러한 과정을 통해서 압수수색 현장에서 은닉행위가 발생할 경우 제시한 조사 방법과 분류한 행위 유형에 따라 디지털포렌식 수사관의 신속하고 정확한 판단을 할 수 있도록 함으로써 압수수색 진행의 전반적인 과정의 효율성을 높였다.

하지만 피 압수자의 은닉행위를 판단하기 위해서는 시스템 아티팩츠의 분석뿐만 아니라, 전산관리자가 작성한 문서 등을 종합적 검토하여야 한다. 또한 각 시스템 아티팩츠는 별도의 환경 설정 값에 따라 생성 및 보관주기가 결정되기 때문에 반드시 존재하는 것은 아니다.

향후 본 논문에서 다루지 않은 시스템에 대한 아티팩츠에 대한 연구 및 로그파일 등을 별도에 서버나 데이터베이스의 저장하는 형태에 대한 대처 방법에 대한 연구가 필요하다.

참 고 문 헌

1. Paul M. Wright, Oracle Database Forensics using LogMiner, SANS London June 2004 Conference(2005)
2. Paul S. Randal, Understanding Logging and Recovery in SQL Server, Microsoft TechNet Magazine(2009. 2.)
3. 박수영, 데이터베이스 내 삭제된 레코드의 복원 방법에 관한 연구, 고려대학교(2013)
4. 신경준, 사이버 침해사고 유형별 디지털포렌식 증거의 식별 및 수집에 관한 연구, 고려대학교(2007)
5. 김성혜 외 3명, 분석 목적별 분류기반의 데이터베이스 포렌식 모델, 고려대학교(2008)
6. Harmeet Kaur Khanuja and D.S.Adane, A FRAMEWORK FOR DA TABASE FORENSIC ANALYSIS, Computer Science & Engineering: An International Journal (CSEIJ), Vol.2,(2012)
7. WIKIPEDIA, Usage_share_of_operating_systems, https://en.wikipedia.org/wiki/Usage_share_of_operating_systems
8. WIKIPEDIA, 데이터베이스 관리 시스템, https://ko.wikipedia.org/wiki/데이터베이스_관리_시스템
9. Oracle, Oracle Database 11g: Administration Workshop 1, 188-192
10. Kevvie Fowler, SQL SERVER forensic analysis, Addison-Wesley (2009)
11. Apache, 로그파일, https://httpd.apache.org/docs/2.2/ko/logs.html
12. MS TechNet, IIS에서 로깅 구성, https://technet.microsoft.com/ko-kr/library/hh831775(v=ws.11).aspx

13. WikiKernel, Ext4 Disk Layout, https://ext4.wiki.kernel.org/index.php/Ext4_Disk_Layout#The_Super_Block

Abstract

A Study on System Artifacts Investigation Technique for Identifying Database Concealment

Jin Jong Min
Department of Mathmatical Information Science,
Digital forensic
The Graduate School
Seoul National University

In the course of enforcing a search warrant search, the digital forensic investigator faces a variety of situations. Accordingly, it is necessary to promptly judge the situation and respond appropriately. Judgment and response of the investigator to the situation of the site greatly affects the result of the seizure search. In recent on scene of enforcing a search warrant, it is not uncommon to delete data related to crime allegations stored in the database or to hide the contents of

a production medium or file where data is stored.

In the case of such database concealment, it is necessary to review various aspects from the legal, technical and procedural point of view at the seizure search site. First, from a legal point of view, there is a need for judgment on the seizure of whole data files, including those not related to allegations. In addition to the fact of the crime of this case, additional evidence is required to obtain evidence of the alleged extinction. In order to identify concealment activities, we examine and collect artifacts that are generated by the database and its associated systems. However, it is practically impossible to investigate all the artifacts generated by the system in order to analyze the behavior of the user. Therefore, among many artifacts, artifacts related to concealment of the database should be defined in advance. In addition, specific plans or procedures are needed to efficiently carry out the whole process of seizure search, reflecting the above legal and technical considerations.

This paper suggests method to identify such concealment activities through investigations of system artifacts when the confiscated data hides the data stored in the database related to the alleged crime. The system artifacts that are generated by the concealed actions performed by the concealed actors for each type are defined and classified into three types of the concealment of the confiscated persons: hard disk replacement, database replacement, and record

deletion. Through this process, if the concealment occurs at the seizure search site, the efficiency of the seizure search process is enhanced by conducting the investigation according to the behavior type and the seizure procedure.

keywords : Digital Forensics, Database, Concealment, Artifacts, Investigation Methods, Procedures

Student Number **: 2015-26067**